Facultad 4

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

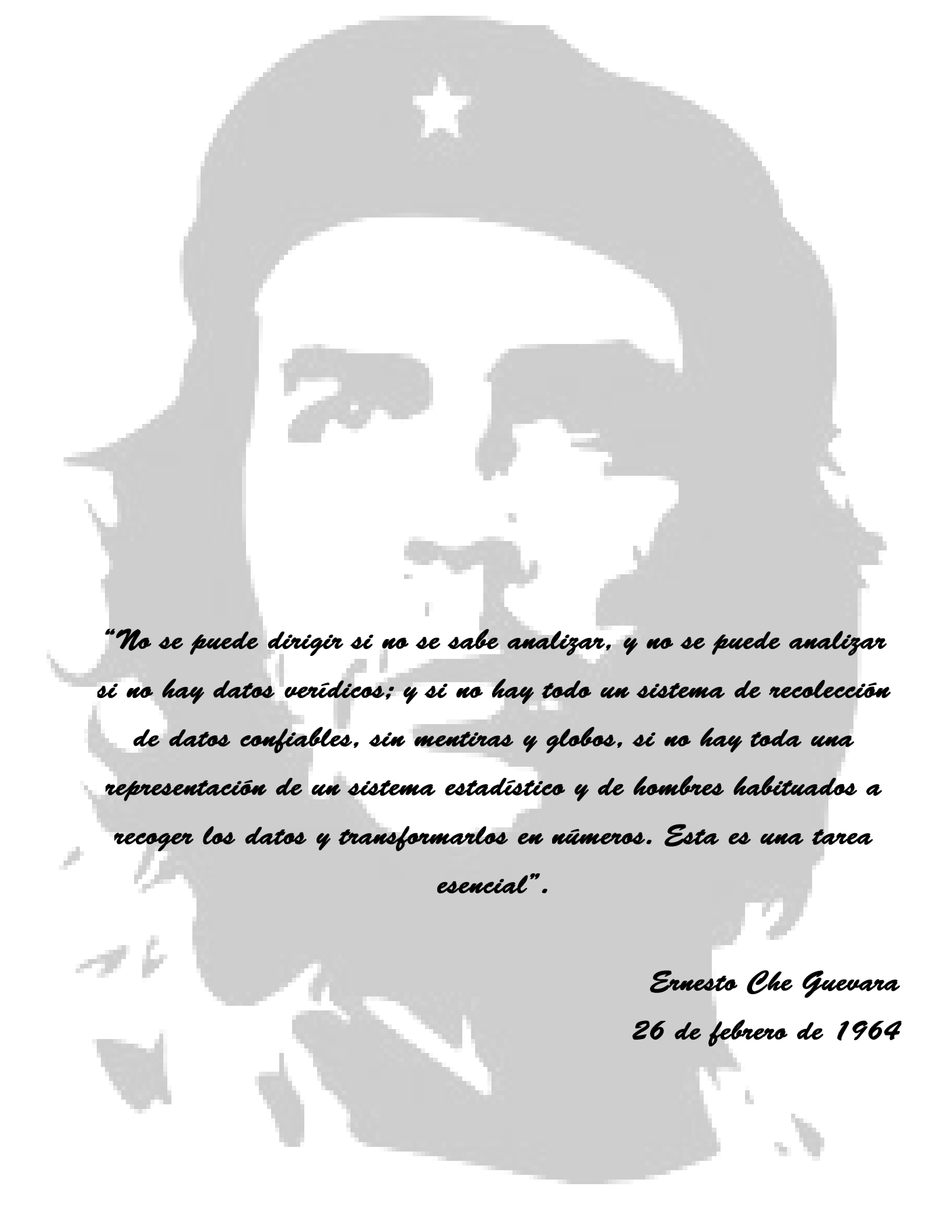
Almacén de datos para el Entorno Virtual de Aprendizaje.

Autor: Ovidio José Castellón Martín.

Tutor: Ing. Eduardo Alfonso Sánchez.

La Habana, 17 de Junio de 2014

"Año 56 de la Revolución"



"No se puede dirigir si no se sabe analizar, y no se puede analizar si no hay datos verídicos; y si no hay todo un sistema de recolección de datos confiables, sin mentiras y globos, si no hay toda una representación de un sistema estadístico y de hombres habituados a recoger los datos y transformarlos en números. Esta es una tarea esencial".

Ernesto Che Guevara

26 de febrero de 1964

Dedicatoria:

A mi madre querida, la luz de mi vida, por su ternura, amor y apoyo. Eres mi mayor inspiración, mi fuerza, ejemplo de lucha, a quien le debo lo que soy. Tú fuiste la que me enseñaste que todo en la vida se puede, esto es por ti y para ti. Te amo mi gordi.

A mi padre por apoyar mis decisiones, ayudarme durante todo este período profesional de mi vida, espero te sientas orgulloso de mi. Te amo.

A mi abuela que siempre ha sido ejemplo y guía, por apoyarme y darme tu amor. Te amo.

A mi hermana Liatne por su amor y apoyo incondicional. Te quiero tata.

A mi hermanita Lilié que aunque ahora es muy chiquita pero sé que me quiere como yo la quiero a ella, con sus locuras y travesuras.

A mi novia Orlay por entregarme su amor, dedicación, apoyo, ternura y comprensión durante todo este tiempo, con la que aprendí a vivir y compartir en pareja, a la que me embrujó desde la primera vez que la besé. Siempre te amaré mi bruji.

A la memoria de mi tío Ovidio al que todos le llamábamos Vivi. Siempre te recordaré.

Agradecimientos:

A mis padres por apoyarme en todo, por educarme como lo ha hecho hasta ahora, por aconsejarme siempre y por ayudarme a ser el joven que soy.

A mi abuela por darme su amor y confianza, por sus consejos y ternura.

A mis hermanas Liatne, Lilié, a mis hermanas postizas Geisy y Marisley por su cariño y apoyo.

A mis tíos Vivi, Marcial, el Chino y Pando por todo su apoyo y confianza.

A mis tías Juana, Victoria, Nury, Maribel y Malbin por su amor, apoyo y dedicación.

A mis segundas madres de corazón a Maribel, Mercedes, Aleida, Mayda, mi yaya Mayely, por su dedicación y amor.

A mis primos, que más que primos, son mis hermanos Yhasmany, Marcialito, Juancarlito, Yordany.

A mis primas Rosmary, Yilian, Greter, por su cariño y apoyo.

A mi novia Orlay y a su familia que ya es mía también, a Margot, Marta, Eliodoro, Enrique, Erlis, Edit, Marta Celia, y a todos por aceptarme como uno más de su familia.

A mi familia de corazón en Varadero a Julia, al Negro, Suyin, Marrero y a Mauricio, por entregarme su cariño y aceptarme como parte de su familia.

A mi familia de corazón Perla, Juan C, Daniel, Nela y Carlitos por su apoyo y cariño.

A toda mi familia porque todos han contribuido en mi formación desde que era un niño. Todos me han apoyado, me han ayudado, me han aconsejado y han logrado que sea el joven que soy.

A mi amigo Yasiel que venimos juntos desde el politécnico, por soportarme y ayudarme cada vez que lo he necesitado.

A mis amistades Roberto con sus cuentos, locuras y alegría, Malidia con su alegría, Geovelsi con su vos y cariño, Luis T. el que todo lo arregla, Ruby con su cariño, Lislien con su amor y fuerza, Carlos A. con su inteligencia y apoyo, Danis C. con su vos y compañerismo, Victor con sus cuentos y locura, Marielí, Yunieska, Edwin y José L., Yudy, Angel, Yania, Raisel, Ismari, Dani M., Lazaro, Frank, Lorenzo, Marisol, Yubeni, en fin a todos los que estuvieron y compartieron conmigo durante estos 5 años.

A Cachan y su familia por su apoyo, disposición y ayuda.

A todos los que fueron mis profesores durante toda la carrera especialmente a Roberto López, Yinimary, Sandra, Leonardo, Mariatne, Arlenys, Lombillo, Arcel, Osbaldo por transmitirme sus conocimientos y comprensión.

A todas las personas que me ayudaron en la realización de esta investigación especialmente a los profesores Haydee, Yaima, Salvador, Manolo, Marisel.

A mi tutor Eduardo, por su apoyo y ayuda.

A mi oponente Yandris, que con sus señalamientos y preguntas hicieron que mi investigación tuviese mejor calidad.

A los profesores del tribunal, que con sus señalamientos y recomendaciones me permitieron obtener este trabajo como resultado de mi carrera.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ___ días del mes de _____ del año _____.

_____	_____
Ovidio José Castellón Martín	Ing. Eduardo Alfonso Sánchez.
Autor	Tutor

Resumen

Son numerosas las herramientas que generan un cúmulo cada vez mayor de información. Los centros educativos como organizaciones, necesitan perfeccionar la toma de decisiones relacionada a los procesos que desarrollan, por lo que han potenciado la investigación científica como alternativa para intensificar el uso de los datos que gestionan, esto redundará en una ventaja competitiva sólida.

En la presente investigación se desarrolla una propuesta de un Almacén de Datos (AD), dirigida a estructurar de una forma adecuada la información que se genera por la interacción de los usuarios con el Entorno Virtual de Aprendizaje (EVA) de la Universidad de las Ciencias Informáticas (UCI), para el posterior análisis de los datos. Moodle es la representante de los sistemas para la gestión del aprendizaje en la UCI, que desde el año 2005 fue implantado como soporte tecnológico al proceso de enseñanza-aprendizaje. Dicho sistema no posee actualmente una herramienta que permita contribuir a la toma de decisiones en cuanto al uso que tienen los recursos publicados, ni les facilite realizar y visualizar reportes de manera ágil, sencilla y sin interrumpir el desarrollo del entorno. Esta propuesta está fundamentada en el estudio de las metodologías, herramientas y procesos asociados a la construcción de AD.

Palabras clave: almacén de datos, educación virtual, moodle, toma de decisiones, sistemas de gestión de aprendizaje.

Índice

Introducción

Actualmente existe un acelerado desarrollo en la informatización de la sociedad, lo que provoca un crecimiento en la capacidad de generar y almacenar información. Cuanto mayor es la capacidad para almacenar datos, mayor es la dificultad para extraer conocimientos realmente útiles. Las empresas e instituciones se han dado cuenta de la creciente necesidad de procesar dicha información, siendo realmente eficaz si los datos están ordenados, analizados y transformados en tiempo real de modo que permitan resolver problemas específicos (1).

Con la automatización de los datos se puede descubrir información valiosa, la cual se puede aprovechar y convertir en una oportunidad de negocio, utilizando la información extraída y procesada para elaborar efectivos planes de inteligencia de negocio (*Business Intelligence*, BI por sus siglas en inglés). Estos planes logran oportunidades de capitalizar con rapidez las ventajas competitivas de una organización, analizar de manera rápida y sencilla la información para la toma de decisiones a nivel operativo, táctico y estratégico; lo que conlleva a que aparezcan procesos y tecnologías nuevas que buscan suplir las necesidades de manejo de la información existente. Dentro de estas tecnologías surgen nuevos conceptos: los AD, proveen un ambiente para que las organizaciones hagan un mejor uso de los datos que manejan, posibilitan un aumento en su rendimiento, teniendo información real y oportuna lo cual ayuda a la toma de decisiones (1).

El análisis de los datos para la toma de decisiones no se limita a los sectores económicos, con todo este desarrollo tecnológico e informacional han surgido durante la última década campos mucho más amplios: la educación. Al igual que en otros ámbitos, se han implementado nuevos recursos que ponen de manifiesto la necesidad de reconceptualizar los procesos y modelos tradicionales de enseñanza y aprendizaje. Para ayudar al proceso docente se hace necesario la creación y utilización de sistemas informáticos, en el que los estudiantes cada vez más se sensibilicen e incrementen su interés hacia el estudio, para así complementar o presentar alternativas en los procesos de la educación tradicional.

Como alternativa para el proceso de formación se han desarrollado los sistemas de gestión de aprendizaje (*Learning Management System*, LMS por sus siglas en inglés), que comprenden cualquier actividad educativa que utilice medios electrónicos para realizar todo o parte del proceso formativo; permitiendo así la capacitación de manera no presencial, eliminando las barreras de tiempo y distancia en el proceso de

enseñanza-aprendizaje, adecuándose a las habilidades, necesidades y disponibilidades de cada estudiante (2).

La introducción de los LMS ha generado considerables volúmenes de datos. El estudio de las trazas en el proceso educativo puede ser usado para mejorar dinámicamente el desarrollo del análisis del aprendizaje. La idea fundamental consiste en la interpretación de las trazas correspondientes a las acciones de los estudiantes en el progreso de su aprendizaje; para evaluar el desempeño académico, predecir actuaciones futuras e identificar los problemas potenciales. En resumen, contribuir a la toma de decisiones en el proceso de enseñanza-aprendizaje (2).

Desde la implantación de la plataforma Moodle en la UCI, con la creación del EVA, se contó con un módulo que facilita la creación de reportes sobre la participación de los usuarios y el uso que estos le dan a los recursos de aprendizaje. En general estos reportes se basan en mostrar el contenido de los **logs** de una forma tabular. Debido a la dificultad que podría traer obtener información de reportes organizados de esta manera, los usuarios debían obtener los datos, por ejemplo, en archivos Excel y realizar un procesamiento ya independiente de las facilidades que le podría brindar el EVA.

Teniendo en cuenta que las investigaciones realizadas hasta el momento no satisfacen las necesidades de almacenamiento con una estructura adecuada, sobre los datos generados a partir de la interacción de los usuarios con el EVA en la UCI, de una forma ágil, sencilla y que no intervenga en el funcionamiento de la plataforma. Se plantea como **problema de la investigación**: ¿cómo almacenar en una estructura adecuada la información generada por la interacción de los usuarios con el EVA de la UCI para su posterior análisis?

Una vez identificado el problema, el **objeto de estudio** se centra en el proceso de almacenamiento y estructuración de los datos.

Como **campo de acción** los almacenes de datos que contribuyen al almacenamiento con una estructura adecuada de la información que se genera de la interacción de los usuarios con el EVA de la UCI.

Para dar solución al problema de la investigación se define como **objetivo general:**

Desarrollar un almacén de datos para guardar de forma estructurada la información que se genera por la interacción de los usuarios con el EVA de la UCI que permita su posterior análisis.

Desglosándose en los siguientes **objetivos específicos**:

- Efectuar el estudio de las herramientas, metodologías y conceptos comunes en el desarrollo de almacenes de datos.

- Realizar el análisis del AD propuesto.
- Diseñar el AD propuesto.
- Desarrollar el AD de la propuesta.
- Validar la solución propuesta a través de las pruebas de aceptación e integración.

Idea a defender:

El desarrollo de un almacén de datos con una estructura adecuada que unifique la información relacionada a la interacción de los usuarios con el EVA de la UCI, contribuirá al análisis de la misma de una forma ágil, sencilla y que no intervenga en el correcto funcionamiento del sistema.

A lo largo del presente documento se describen los resultados de esta investigación, el cual se estructura por la presente introducción, dos capítulos, conclusiones, recomendaciones y anexos que complementan la información comprendida en ella quedando la estructura capitular de la siguiente manera:

Capítulo 1: Herramientas y metodologías comunes en el desarrollo de almacenes de datos

Se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo de diploma. En este capítulo se realiza un estudio de las soluciones similares, se analizan las metodologías y herramientas que se ajustan al desarrollo de la investigación, justificando la selección y utilización de cada una de ellas.

Capítulo 2: Análisis, diseño, desarrollo y prueba del almacén de datos propuesto

En este capítulo se describe la aplicación de la metodología seleccionada, el desarrollo de los procesos definidos por esta, los artefactos generados, la identificación de los requisitos de información, la especificación de las medidas, hechos, dimensiones y los modelos de datos. Se realiza el desarrollo del subsistema de integración de datos, donde se explica los procesos de extracción, transformación y carga de los datos. Se demuestra el proceso de pruebas del sistema, para el cuál se utilizaron las pruebas de integración y aceptación que permitirán evaluar el nivel de la calidad de los datos y poder comprobar que se satisfacen las necesidades del cliente.

Capítulo 1: Herramientas y metodologías comunes en el desarrollo de almacenes de datos

Este capítulo abarca todos los elementos teóricos que sustentan el objeto de estudio y el campo de acción de la investigación. Se describen herramientas y soluciones similares que logran un eficiente manejo de la información en sistemas con grandes volúmenes de datos, así como diferentes metodologías para el desarrollo de almacenes de datos. Se relacionan conceptos que desde el punto de vista teórico permiten un mejor entendimiento de lo que se plantea en la situación problemática. Finalmente se argumenta la selección de la metodología y tecnologías que son empleadas en función del cumplimiento satisfactorio del objetivo propuesto en este trabajo de diploma.

1.1 Soluciones existentes

En el mundo existen múltiples instituciones que utilizan tecnologías informáticas que posibilitan el análisis de la información generada por la interacción de los usuarios con diferentes plataformas educativas. Después de haber estudiado la bibliografía se encontraron algunas investigaciones que utilizaron los almacenes de datos como tecnología de apoyo, para almacenar de una forma estructurada y posteriormente analizar los datos, algunos de estos son:

- El almacén de datos para una herramienta de seguimiento de usuarios de la Universidad Católica de Loja, el cual permite estructurar y conocer información sobre la interacción de los usuarios sobre la plataforma Moodle (2).
- El almacén de datos de la Plataforma Educativa ZERA desarrollada en el centro FORTES perteneciente a la Facultad 4 de la (UCI), que permite estructurar y analizar la información generada por la misma (1).
- El almacén de datos de la plataforma Desire2Learn en la universidad de Wisconsin, para investigar el potencial y las capacidades del sistema de éxito estudiantil (3).

Luego de la revisión y análisis las potencialidades de los almacenes de datos en estas investigaciones, cada una con características, facilidades y objetivos diferentes, se decide desarrollar un almacén de datos que contribuya al análisis de la información que se genera por la interacción de los usuarios con el EVA de la UCI.

1.2 E-Learning

El reconocido autor Miguel Varela, prestigioso por la excelencia de sus publicaciones sobre el tema, define el e-learning como: "*La educación a distancia completamente virtualizada a través de los nuevos canales*

electrónicos, utilizando para ello herramientas o aplicaciones de hipertexto como correo electrónico, páginas web, foros de discusión, mensajería instantánea y plataformas de formación sirviendo de soporte en los procesos de enseñanza aprendizaje" (4).

Según Tanino Ferri, e-learning "… *es una manera flexible y poderosa mediante la cual individuos y grupos adquieren nuevos conocimientos y destrezas con apoyo de tecnología de redes de computadoras. Permite diseminar y tener acceso a información multimedia, hacer uso de simuladores, al tiempo que permite interacción y colaboración con personas interesadas que pueden encontrarse dispersas alrededor del mundo"* (4).

El Ing. Robinson Martínez plantea que: "*El e-learning es un modelo de formación a distancia que utiliza Internet como herramienta de aprendizaje. Este modelo permite al alumno realizar el curso desde cualquier parte del mundo y a cualquier hora"* (4).

En resumen, se puede inferir que e-learning no es más que un modelo de formación que utiliza las Tecnologías de la Información y las Comunicaciones (TIC) como intermediaria y protagonista para que de una forma u otra los conocimientos sean adquiridos por los estudiantes.

1.2.1 Moodle

El Sistema Gestor de Aprendizaje (SGA) Moodle (del inglés *Modular Object – Oriented Dynamic Learning Environment*) es una plataforma para la creación de cursos y sitios web basados en Internet. Su misión consiste en desarrollar nuevas teorías educativas basadas en una gama de recursos didácticos disponibles en ella. Este LMS (*Learning Management System*) es una herramienta potente y sencilla que otorga gran libertad y autonomía en la creación y gestión de cursos. Es utilizado en organizaciones educativas para los procesos de enseñanza, pues fomentan el auto aprendizaje y el aprendizaje colaborativo, la realización de exámenes y evaluación de tareas en línea (5).

Esta plataforma de teleformación se distribuye libremente y de forma gratuita bajo la licencia pública GNU. Moodle está desarrollada en el lenguaje de programación PHP y es multiplataforma. Utiliza la biblioteca ADOdb de abstracción de bases de datos, la cual da soporte a varios tipos de bases de datos, esencialmente MySQL y PostgreSQL (5).

Cuenta con variedad de módulos para la creación de cursos, incluyendo recursos y actividades. Se definen como recursos aquellos que se utilizan en cuestiones de auto-estudio, y priorizan la interacción persona-contenido. Sin embargo las actividades son de tipo colaborativo, las cuales priorizan la

interacción persona(s)-persona(s). Como parte de estos módulos se definen lecciones, tareas, cuestionarios, encuestas, libros, glosarios, wikis, foros, chats, enlaces y etiquetas; con estructuras diferentes según sus objetivos específicos (6).

En resumen, Moodle es una aplicación web de tipo ambiente educativo virtual. Es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. En la UCI es utilizada para la gestión del EVA.

Base de Datos de Moodle

La base de datos de Moodle en su versión 2.3.4 está conformada por 299 tablas, la mayoría de ellas con más de 5 columnas. Dentro de las tablas más importantes para el presente trabajo de diploma se encontraron las siguientes.

- *mdl_log*: Tabla que almacena los log del sistema, es decir, toda la actividad que se realiza sobre la plataforma. En esta tabla no se guarda información específica sobre la interacción de los usuarios con módulos específicos como pueden ser: foros, chat, cuestionarios, etc.
- *mdl_user*: Tabla que almacena los usuarios del sistema.
- *mdl_role*: Tabla que almacena los roles del sistema. Por ejemplo: estudiante, profesor, administrador.
- *mdl_role_assignments*: Tabla que almacena la relación entre los usuarios y los roles del sistema.
- *mdl_course*: Tabla que almacena los cursos del sistema.
- *mdl_course_modules*: Tabla que almacena la relación entre los cursos y los módulos.
- *mdl_groups*: Tabla que almacena los grupos que se encuentran conformados en el sistema.
- *mdl_groups_members*: Tabla que almacena la relación entre los grupos y usuarios.

1.3 Almacén de Datos

Según Ralph Kimball, reconocido autor en el tema de almacenes de datos, los define como una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis; es la unión de todos los MD de una entidad" (7).

Para Cesares son una colección de datos orientados a temas, integrados, no volátil, de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales (8).

El autor Bill Inmon los define como una colección de datos orientados a temas, integrados, no volátiles e historiados, organizados para el apoyo de un proceso de ayuda a la decisión" (9).

Se concluye que, los almacenes de datos (AD) son una colección de datos orientados a temas, integrados, variables en el tiempo y no volátiles, donde su principal función es contribuir a la toma de decisiones basado en la información histórica de las instituciones.

1.3.1 Características principales

Precisamente es Inmon quien definió las características principales de los AD. Estas son:

Orientados al tema: La información se clasifica en base a los aspectos que son de interés para la empresa. Siendo así, los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones.

Integrados: Los datos deben de ser consistentes siempre dentro del almacén de datos e integrados de distintas fuentes de datos operacionales. Algunos ejemplos de integración de los datos son:

- Medida de atributos: Los diseñadores de aplicaciones miden las unidades de medida en una variedad de formas. Un diseñador almacena los datos de longitud en centímetros, otros en pulgadas, otros en metros y otros en kilómetros. Al dar medidas a los atributos, la transformación traduce las diversas unidades de medida usadas para transformarlas en una medida estándar común.
- Fuentes Múltiples: El mismo elemento puede derivarse desde fuentes múltiples. En este caso, el proceso de transformación debe asegurar que la fuente apropiada sea usada, documentada y movida al almacén de datos.

No volátiles: La manipulación de datos en el almacén de datos es mucho más simple que en el ambiente operacional. Hay dos únicos tipos de acciones: la carga inicial de datos y el acceso a los mismos. Los datos almacenados no se modifican ni se eliminan nunca, solo se añaden nuevos datos.

De tiempo variante: Toda la información del almacén de datos es requerida en algún momento. Esta característica básica de los almacenes de datos, es muy diferente de la información encontrada en el ambiente operacional. En éstos, cuando se accede a una unidad de información, se espera que los valores requeridos se obtengan a partir del momento de acceso. Como la información en el almacén de datos es solicitada en cualquier momento, los datos encontrados en el depósito se llaman de tiempo variante.

Las variantes del tiempo se pueden notar de tres formas:

- Límite de tiempo: El margen de tiempo del almacén de datos es mucho mayor en cuanto a los datos (puede contener datos entre 5 y 10 años de almacenamiento). Por otro lado, en el ambiente operacional, el margen de tiempo de almacenamiento de los datos es mucho menor (contiene datos entre 60 y 90 días); ya que un programa de aplicación para trabajar eficientemente debe llevar la mínima cantidad de datos necesarios para realizar las transacciones.
- Clave de estructura: Los datos en el almacén de datos contienen un elemento de tiempo (día, semana, mes y año).
- Actualizaciones: Los datos una vez almacenados correctamente en el almacén de datos no pueden ser alterados, por lo tanto no se pueden actualizar.(10)

Estas características de los almacenes de datos hacen más fácil el acceso de los usuarios finales a una gran variedad de datos. Su empleo en sistemas, facilita el apoyo de toma de decisiones; lo que aumenta el valor operacional de las aplicaciones empresariales, en especial la gestión de relaciones con clientes.

1.3.2 Ventajas del uso de los almacenes de datos

En este epígrafe de la investigación se presentan las principales ventajas de los AD:

- Transforman datos orientados a las aplicaciones en información orientada a la toma de decisiones.
- Integran y consolidan diferentes fuentes de datos en una única plataforma sólida y centralizada.
- Proveen las capacidades de analizar y explotar toda la información que poseen.
- Permiten reaccionar rápidamente a los cambios del mercado.
- Aumentan la competitividad en el mercado.
- Mejoran la entrega de información, es decir, información completa, correcta, consistente, oportuna y accesible.
- Facilitan la aplicación de técnicas estadísticas de análisis y modelización para encontrar relaciones ocultas entre los datos del almacén; obteniendo un valor añadido para el negocio de dicha información.
- Los usuarios pueden tener a su disposición una gran cantidad de información multidimensional, presentada coherentemente como fuente única, confiable y disponible en sus estaciones de trabajo.

- Proporcionan la capacidad de aprender de los datos del pasado y predecir situaciones futuras en diversos escenarios. (10)

1.3.3 Proceso de construcción y población del almacén de datos

En el proceso de construcción y población de un almacén de datos, se relacionan diferentes conceptos, que permiten separar y estructurar las diversas actividades a realizar en su desarrollo, dando más profesionalidad y calidad a la investigación.

1.3.3.1 Modelo Multidimensional

Una base de datos multidimensional es donde su información se almacena en forma multidimensional, es decir, a través de tablas de hechos y tablas de dimensiones (11).

Proveen una estructura que permite, a través de la creación y consulta a una estructura de datos determinada (cubo multidimensional, *Business Model*, etc), tener acceso flexible a los datos, para explorar y analizar sus relaciones, y consiguientes resultados (11).

Las bases de datos multidimensionales implican tres variantes posibles de modelamiento, que permiten realizar consultas de soporte de decisión:

- Esquema en Estrella (*Star Scheme*).

Un esquema en estrella es un modelo de datos que tiene una tabla de hechos que contiene los datos para el análisis, rodeada de las tablas de dimensiones. Este aspecto, de tabla de hechos (o central) más grande rodeada de radios o tablas más pequeñas es lo que asemeja a una estrella, dándole nombre a este tipo de construcciones. Las tablas de dimensiones tendrán siempre una clave primaria simple, mientras que en la tabla de hechos, la clave principal estará compuesta por las claves principales de las tablas dimensionales (12).

Este esquema tiene características que a la vez son sus ventajas, algunas de estas son:

- Posee los mejores tiempos de respuesta.
- Su diseño es fácilmente modificable.
- Existe paralelismo entre su diseño y la forma en que los usuarios visualizan y manipulan los datos.
- Simplifica el análisis.
- Facilita la interacción con herramientas de consulta y análisis. (12)

- Esquema Copo de Nieve (*Snowflake Scheme*).

Este esquema representa una extensión del modelo en estrella cuando las tablas de dimensiones se organizan en jerarquías de dimensiones.

Existe una tabla de hechos central que está relacionada con una o más tablas de dimensiones, quienes a su vez pueden estar relacionadas o no con una o más tablas de dimensiones. Este modelo es más cercano a un modelo de entidad relación, que al modelo en estrella, debido a que sus tablas de dimensiones están normalizadas (12).

Una de los motivos principales de utilizar este tipo de modelo, es la posibilidad de segregar los datos de las tablas de dimensiones y proveer un esquema que sustente los requerimientos de diseño. Otra razón es que es muy flexible y puede implementarse después de que se haya desarrollado un esquema en estrella (12).

Se pueden definir las siguientes características de este tipo de modelo:

- Posee mayor complejidad en su estructura.
- Hace una mejor utilización del espacio.
- Es muy útil en tablas de dimensiones de muchas tuplas.
- Las tablas de dimensiones están normalizadas, por lo que requiere menos esfuerzo de diseño.
- Puede desarrollar clases de jerarquías fuera de las tablas de dimensiones, que permiten realizar análisis de lo general a lo detallado y viceversa. (12)

A pesar de todas las características y ventajas que trae aparejada la implementación del esquema copo de nieve, existen dos grandes inconvenientes de ello:

- Si se poseen múltiples tablas de dimensiones, cada una de ellas con varias jerarquías, se creará un número de tablas bastante considerable, que pueden llegar al punto de ser inmanejables.
- Al existir muchas uniones y relaciones entre tablas, el desempeño puede verse reducido. (12)

- Esquema Constelación o copo de estrellas (*Starflake Scheme*).

Su diseño y cualidades son muy similares a las del esquema en estrella, pero posee una serie de diferencias con el mismo, que son precisamente las que lo destacan y caracterizan. Entre ellas se pueden mencionar:

- Permite tener más de una tabla de hechos, por lo cual se podrán analizar más aspectos claves del negocio con un mínimo esfuerzo adicional de diseño.

- Contribuye a la reutilización de las tablas de dimensiones, ya que una misma tabla de dimensión puede utilizarse para varias tablas de hechos.
- Posibilidad de navegar de un hecho hacia otro y la optimización del espacio gracias a la compartición de dimensiones, evitando así la redundancia de datos. (12)

Los mencionados esquemas pueden ser implementados de diversas maneras, que, independientemente al tipo de arquitectura, requieren que toda la estructura de datos esté desnormalizada o semidesnormalizada, para evitar desarrollar uniones (Join) complejas para acceder a la información, con el fin de agilizar la ejecución de consultas. Los diferentes tipos de implementación son los siguientes:

- Relacional - ROLAP.
- Multidimensional - MOLAP.
- Híbrido - HOLAP.(11)

La construcción de herramientas ROLAP sobre sistemas relacionales permiten más escalabilidad para manejar grandes volúmenes de datos, especialmente modelos con dimensiones de gran cardinalidad. Los tiempos de carga son generalmente mucho menores que con las cargas MOLAP automatizadas. Obviando el almacenamiento de datos del modelo multidimensional, es posible modelar datos con éxito que de otro modo no se ajustarían en un modelo dimensional estricto.

1.3.3.2 Procesamiento de Transacciones en Línea

Un Procesamiento de Transacciones en Línea (OLTP por sus siglas en inglés, *Online Transaction Proccesing*), representa toda aquella información transaccional que genera la empresa en su accionar diario, además, de las fuentes externas con las que puede llegar a disponer. Estas fuentes de información, son de características muy disímiles entre sí, en formato, procedencia, función, etc.

Los sistemas OLTP generalmente guardan la información en:

- Archivos de textos.
- Hipertextos.
- Hojas de cálculos.
- Informes semanales, mensuales, anuales, etc.
- Bases de datos transaccionales.(13)

El procesamiento de transacciones en línea cada vez necesita más recursos para las transacciones que se propagan por una red y que pueden integrar a más de una empresa. Por esta razón, el software actual

para sistemas OLTP utiliza procesamiento cliente-servidor y software de intermediación (middleware) que permite a las transacciones correr en diferentes plataformas en una red.

1.3.3.3 Procesamiento Analítico en Línea

El término Procesamiento Analítico en Línea (OLAP por sus siglas en inglés, *Online Analitical Proccesing*), define a una tecnología que se basa en el análisis multidimensional de los datos y que le permite al usuario tener una visión más rápida e interactiva de los mismos.

Las herramientas OLAP ofrecen:

- Una visión multidimensional de los datos (matricial).
- No imponer restricciones sobre el número de dimensiones.
- Ofrecer simetría para las dimensiones, permitir definir de forma flexible (sin limitaciones) sobre las dimensiones: restricciones, agregaciones y jerarquías entre ellas.
- Ofrecer operadores intuitivos de manipulación: *dril down, roll up, slice and dice, pivot*.
- Ser transparentes al tipo de tecnología que soporta el almacén de datos (ROLAP o MOLAP).(14)

La razón de usar OLAP para las consultas es la rapidez de respuesta. Una base de datos relacional almacena entidades en tablas discretas si han sido normalizadas. Esta estructura es buena en un sistema OLTP pero para las complejas consultas que relacionan varias tablas es relativamente lenta. Un modelo mejor para búsquedas (aunque peor desde el punto de vista operativo) es una base de datos multidimensional.

1.3.3.4 Mercado de datos (MD)

Subconjunto de la información de un AD, generalmente de un solo proceso de negocio, que se dirige a un determinado departamento/grupo de usuarios. Normalmente contiene la información de un diagrama en estrella por lo que se suelen utilizar como sinónimos, aunque conceptualmente son diferentes. Descomponer el AD en diferentes MD suele mejorar el rendimiento de las consultas al reducir el volumen de datos que se recorren para responder.

Los MD se utilizan para:

- Segmentar la información en diferentes plataformas de hardware (posible portabilidad).
- Facilitar el acceso de las herramientas de consulta.
- Dividir los datos para controlar mejor los accesos.

- Mejorar los tiempos de respuesta.

Los MD pueden ser necesarios en control de accesos:

- En los Sistemas Gestores de Bases de Datos (SGBD) tradicionales sólo permiten restringir acceso a tablas, no a filas.
- Con un MD se pueden separar físicamente porciones completas de datos.(14)

Es además un modelo multidimensional basado en tecnología OLAP que representa a un área específica de la empresa, incluyendo las variables claves y los indicadores para el proceso de toma de decisiones.

1.3.3.5 Procesos de extracción, transformación y carga (ETL)

Extraer, Transformar y Cargar (ETL por sus las siglas en inglés, *Extract, Transform and Load*). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, MD, o AD para analizar, o en otro sistema operacional para apoyar un proceso de negocio.(15)

Proceso de extracción:

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

Una parte intrínseca del proceso de extracción es la de analizar los datos extraídos, de lo que resulta un chequeo que verifica si los datos cumplen la pauta o estructura que se esperaba. De no ser así los datos son rechazados.

Proceso de transformación:

La fase de transformación de un proceso de ETL aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos. No obstante en otros casos pueden ser necesarias aplicar algunas de las siguientes transformaciones:

- Seleccionar sólo ciertas columnas para su carga (por ejemplo, que las columnas con valores nulos no se carguen).
- Traducir códigos (por ejemplo, si la fuente almacena una “H” para Hombre y “M” para Mujer pero el destino tiene que guardar “1” para Hombre y “2” para Mujer).
- Codificar valores libres (por ejemplo, convertir “Hombre” en “H” o “Sr” en “1”).
- Obtener nuevos valores calculados (por ejemplo, total_venta = cantidad * precio).
- Unir datos de múltiples fuentes (por ejemplo, búsquedas, combinaciones, entre otros).
- Calcular totales de múltiples filas de datos (por ejemplo, ventas totales de cada región).
- Generación de campos clave en el destino.
- Transponer o pivotar (girando múltiples columnas en filas o viceversa).
- Dividir una columna en varias (por ejemplo, columna “Nombre: García, Miguel”; pasar a dos columnas “Nombre: Miguel” y “Apellido: García”).
- La aplicación de cualquier forma, simple o compleja, valida los de datos ejecutando las acciones que en cada caso se requiera:
    - Datos OK: Entregar datos a la siguiente etapa (Carga).
    - Datos erróneos: Ejecutar políticas de tratamiento de excepciones (por ejemplo, rechazar el registro completo, dar al campo erróneo un valor nulo o un valor centinela).(15)

Proceso de carga:

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los AD mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.
Existen dos formas básicas de desarrollar el proceso de carga:

- Acumulación simple: La acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el AD, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada.

- *Rolling*: El proceso de *Rolling* por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (por ejemplo, totales diarios, totales semanales, totales mensuales, etc.).(15)

Los procesos ETL son los componentes más importantes y de mayor valor añadido en una infraestructura que implique la integración de varias fuentes de datos. En consecuencia, representan un pilar fundamental tanto de simples proyectos de recopilación como de soluciones complejas de BI, especialmente si se requiere mucha precisión o actualización en los datos.

1.4 Metodologías para el diseño e implementación de un almacén de datos

Una metodología es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Actualmente existen una gran variedad metodologías que definen y guían el ciclo de vida de desarrollo de la solución. Las tendencias más conocidas son las de Bill Inmon y Ralph Kimball, que pretenden dar un acercamiento a una propuesta ideal para el desarrollo de almacenes de datos, las cuales no son las únicas. Se han desarrollado otras que no siguen específicamente uno de los enfoques de los autores anteriores, sino que realizan una selección de lo mejor de cada uno y definen su propia metodología.

1.4.1 Metodología de Kimball

La metodología Kimball, creada por Ralph Kimball, es la metodología por excelencia para los proyectos de almacenes de datos. Se enfoca principalmente en el diseño de bases de datos que almacenarán la información para la toma de decisiones. El diseño se basa en la creación de tablas de hechos que son tablas que contienen información numérica de los indicadores a analizar, es decir la parte cuantitativa de la información.

El método Kimball es iterativo en el cual el AD es construido pieza a pieza. Kimball identifica dimensiones compartidas o comunes (cliente, tiempo, geografía) que van a ser utilizadas para construir múltiples grupos de hechos desde un alto nivel. Sugiere que se construya un grupo de hechos cada vez. Kimball denomina a cada uno de estos grupos "mercado de datos". De esta forma concibe la construcción de un AD mediante el desarrollo de todos los "mercado de datos" y la consecuente población de las dimensiones compartidas. Además, este autor establece 4 fases para el cumplimiento del proyecto:

Fase I Requerimiento y Gestión del Proyecto

- Definición del proyecto.
- Planeación y gestión del proyecto.
- Definición de los requerimientos del usuario.

Fase II Arquitectura

- Diseño Técnico de la Arquitectura.
- Medidas Tácticas de Seguridad.
- Plan Estratégico de Seguridad.
- Selección e Instalación de Productos.

Fase III Diseño e Implementación.

- Análisis Multidimensional (Lógico y Físico).
- Análisis de Fuentes de Datos.
- Diseño & Implementación del Área Temporal.
- Popular& Validar Base de Datos.
- Optimización del Rendimiento
- Especificación y Desarrollo de Aplicaciones de Usuario Final.

Fase IV Implantación & Acciones

- Plan de Implantación.
- Pruebas.
- Implantación.
- Optimización del Rendimiento.
- Mantenimiento.
- Crecimiento.
- Capacitación y Transferencia Tecnológica.(16)

1.4.2 Metodología de desarrollo de AD para Centro de Tecnologías de Gestión de Datos (DATEC)

Esta metodología utiliza el modelo incremental y el modelo de Desarrollo Rápido de Aplicaciones (DRA). El ciclo de vida de la metodología está organizado por fases, algunas de ellas podrán ser implementadas de forma paralela según el componente que se está desarrollando, los componentes se integran a medida que avanza el desarrollo de la solución. Esto permite agilizar la producción, reduciendo los tiempos de

desarrollo. También permite ajustar la metodología al modelo que sigue el centro basado en líneas de producto.

Fases del ciclo de vida

Las fases se definieron teniendo en cuenta las propuestas por la metodología de Kimball, los procesos y actividades presentados anteriormente y las características del desarrollo de proyectos de software en la UCI. Finalmente se obtuvieron las fases que se describen a continuación.

Estudio preliminar y Planeación:

Se realiza un estudio minucioso en la entidad cliente. Esto incluye un diagnóstico de información, de datos y de infraestructura tecnológica, todo esto con el fin de determinar qué es lo que se desea construir y qué condiciones existen para el desarrollo y montaje de la misma. También se llevan a cabo las tareas de planeación del proyecto, se definen los objetivos, el alcance preliminar, los costos estimados, los recursos necesarios, y otras series de actividades.

Levantamiento de requisitos:

Se realiza en tres direcciones, 1ra. Identificación de las metas y objetivos de la organización, 2da. Identificación de las necesidades de información de los clientes y las reglas de negocio; y 3ra. Haciendo un levantamiento detallado de cada una de las fuentes de datos a integrar para validar la disponibilidad de la información.

Arquitectura:

Se define la arquitectura de la solución, aspectos como, la seguridad del sistema, la comunicación entre los subsistemas, la tecnología a utilizar, *hardware* y *software*, entre otros aspectos de gran importancia. Vale aclarar que esta fase puede desarrollarse en paralelo con la fase de Levantamiento de requisitos, siempre y cuando los resultados del diagnóstico tecnológico realizado durante la fase de Estudio preliminar dejen bien definidas las características técnicas de la organización y el cliente sepa lo que desea.

Diseño e Implantación:

Se define el diseño de las estructuras de almacenamiento, se diseñan los procesos de integración de datos como, las reglas de extracción, transformación y carga, se diseñan los cubos para la presentación de los datos, así como el diseño visual de la aplicación definido por el cliente. Después se implementan cada uno de los subsistemas (repositorio de datos, integración de datos, presentación de datos).

Prueba:

Aquí se realizan varias pruebas, comenzando por las Pruebas de Unidad llevadas a cabo por los propios desarrolladores de cada uno de los grupos, luego las Pruebas de Integración y Sistema, hasta llegar a las Pruebas de Aceptación con el cliente final.

Despliegue:

Consta de dos etapas, la primera es un Despliegue Piloto, donde se configuran los servidores necesarios y se instalan las herramientas según la Arquitectura definida, se cargan una muestra de los datos en un ambiente controlado, con el fin de demostrarle al cliente final que la solución funciona. Una vez aceptada la solución por el cliente, se realiza la carga histórica de los datos. Es aquí el momento más idóneo para llevar a cabo la Capacitación y Transferencia Tecnológica a los clientes.

Soporte y Mantenimiento:

Comienza cuando la solución está implantada y en explotación, y se ejecuta según el contrato firmado y las condiciones de soporte establecidas. Puede realizarse a través de varia dos servicios, que pueden ser soporte en línea, vía telefónica, web, correo u otros, hasta el acompañamiento junto al cliente.

Gestión y administración del proyecto:

Esta fase se ejecuta a lo largo de todo el ciclo de vida del proyecto. Es aquí donde se controla, gestiona y chequea todo el desarrollo, los gastos, las utilidades, los recursos, las adquisiciones, los planes y cronogramas entre otras actividades relacionadas con la gestión y administración de proyecto.

1.4.3 Metodología Hefesto

Hefesto es una metodología cuya propuesta está fundamentada en una amplia investigación, comparación de metodologías existentes y experiencias propias en procesos de desarrollo de almacenes de datos. Está orientada a la construcción de almacenes de datos para análisis dimensional (13). Dentro de sus características más comunes encontramos las siguientes:

- Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- Se basa en los requisitos de los usuarios, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.
- Reduce la resistencia al cambio, ya que involucra a los usuarios finales en cada etapa para que tome decisiones respecto al comportamiento y funciones del AD.
- Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.
- Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- Es independiente de las herramientas que se utilicen para su implementación.
- Es independiente de las estructuras físicas que contenga el almacén de datos y de su respectiva distribución.
- Cuando se culmina con una fase, los resultados obtenidos se convierten en el punto de partida para llevar a cabo el paso siguiente.
- Se aplica tanto para AD como para MD. (13)

Esta metodología puede resumirse en las siguientes fases:

Análisis de requerimientos:

Lo primero que se hará es identificar los requerimientos de los usuarios a través de preguntas que expliquen los objetivos de su organización. Luego, se analizarán estas preguntas a fin de identificar cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del almacén de datos. Finalmente se confeccionará un modelo conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.(13)

Análisis de los OLTP:

Se analizarán las fuentes OLTP para determinar cómo serán calculados los indicadores y para establecer las respectivas correspondencias entre el modelo conceptual creado en el paso anterior y las fuentes de datos. Luego, se definirán qué campos se incluirán en cada perspectiva. Finalmente, se ampliará el modelo conceptual con la información obtenida en este paso.(13)

Modelo lógico del Almacén de Datos:

Se confecciona el modelo lógico de la estructura del almacén de datos, teniendo como base el modelo conceptual que ya ha sido creado. Para ello, primero se definirá el tipo de modelo que se utilizará y luego se llevarán a cabo las acciones propias al caso, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.(13)

Proceso ETL:

Una vez construido el modelo lógico, se deberá proceder a probarlo con datos, a través de procesos ETL.
Para realizar la compleja actividad de extraer datos de diferentes fuentes, para luego integrarlos, filtrarlos y depurarlos, existen varios software que facilitan estas tareas, por lo cual este paso se centrará solo en la generación de las sentencias SQL que contendrán los datos que serán de interés.
Antes de realizar la carga de datos, es conveniente efectuar una limpieza de los mismos, para evitar valores faltantes y anómalos.
Al generar los ETL, se debe tener en cuenta cual es la información que se desea almacenar en el depósito de datos, para ello se pueden establecer condiciones adicionales y restricciones. Estas condiciones deben ser analizadas y llevadas a cabo con mucha prudencia para evitar pérdidas de datos importantes.
Cuando se haya cargado en su totalidad el AD, se deben establecer sus políticas de actualización o refresco de datos.(13)

1.4.4 Selección de la metodología

La construcción e implementación de un AD puede adaptarse muy bien a cualquier ciclo de vida de desarrollo de software, con la salvedad de que para algunas fases en particular, las acciones que se han de realizar serán muy diferentes. Se debe tener muy en cuenta, no entrar en la utilización de metodologías que requieran fases extensas de reunión de requerimientos y análisis, fases de desarrollo monolítico que

conlleve demasiado tiempo, ni fases de despliegue muy largas. Por lo que de las metodologías antes expuestas se selecciona la metodología Hefesto ya que con el uso de esta metodología se logra un fácil entendimiento y comprensión de los objetivos y resultados esperados en cada fase, capaz de adaptar con rapidez la estructura ante cualquier cambio en el negocio, involucra a los usuarios finales en cada etapa para que tome decisiones respecto al comportamiento y funciones del AD; es independiente de la herramienta, tipo de ciclo de vida y de la estructura física que se utilicen, además utiliza modelos conceptuales y lógicos sencillos de interpretar y analizar.

1.5 Lenguaje de Modelado Unificado (UML)

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Utilizándolo para el modelado de los artefactos necesarios que garanticen el desarrollo exitoso del AD propuesto.(17)

Se puede aplicar en el desarrollo de software muchas formas para dar soporte a una metodología de desarrollo de software, pero no específica en sí mismo qué metodología o proceso usar.

1.6 Herramientas

Las herramientas, son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. En un sentido amplio del término, se puede decir que una herramienta es cualquier programa o instrucción que facilita una tarea. Dentro del desarrollo del AD, como toda elaboración de un software también existen un grupo de herramientas que facilitarán su construcción.

1.6.1 Herramientas CASE

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas sirven de ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del

código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

1.6.1.1 ArgoUML

Es una herramienta desarrollada en Java que permite crear modelos UML compatibles con los estándares de diferentes versiones de este lenguaje. Incluye una interfaz muy intuitiva, estable y de sencillo manejo. Los tipos de diagramas que se pueden crear son: diagramas de clases, de estados, de actividad, de casos de uso, de colaboración, de despliegue y de secuencia.

- Plataforma: Multiplataforma.
- Licencia: libre BSD (*Berkeley Software Distribution*).
- Extensible: sí
- Código Abierto: sí
- Diagramas que se pueden realizar: de clases, de estado, de actividad, de casos de uso, de colaboración, despliegue, implementación y de secuencia.
- Generación de código: Java, C++, C#, PHP4, PHP5 y SQL.
- Generación de documentación: no.
- Ingeniería inversa: no.
- Exportación de diagramas: GIF, PNG, PostScript, PGML y SVG.
- XMI: sí.
- Ventajas: Genera código automáticamente. Propone soluciones a algunos errores y contiene un panel de propiedades y de tareas pendientes bastante útil. Es una herramienta fácil de usar para el modelado de sistemas.
- Desventajas: ArgoUML está incompleto. No es conforme completamente a los estándares UML y carece de soporte completo para algunos tipos de diagramas incluyendo los diagramas de secuencia y los de colaboración. Otro de sus inconvenientes es que consume muchos recursos. (18)

1.6.1.2 Visual Paradigm

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de

diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE para UML también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML.(19)

Entre sus características fundamentales se encuentran:

Multiplataforma: soportada en plataforma Java para Sistemas Operativos Windows, Linux, Mac OS.

Interoperabilidad: intercambia diagramas UML y modelos con otras herramientas. Soporta la importación y exportación a formatos XMI y XML y archivos excel. Permite importar proyectos de Rational Rose y la integración con Microsoft Office Visio.

Modelado de requisitos: captura de requisitos mediante diagramas de requisitos, modelado de casos de uso y análisis textual.

Colaboración de equipo: realiza el modelado simultáneamente con el Paradigm Team Work Server y Subversión.

Generación de documentación: comparte y genera documentación de diagramas y diseños en formatos PDF, HTML, Microsoft Work.

Editor de detalles de casos de uso: entorno para la especificación de detalles de casos de usos, incluyendo la especificación del modelo general y las descripciones de los casos de uso.

Ingeniería de código: permite la generación de código e ingeniería inversa para los lenguajes: Java, C, C++, PHP, XML, Python, C#, VB .Net, Flash, ActionScript, Delphi y Perl.

Modelado de procesos de negocio: visualiza, comprende y mejora los procesos de negocio con la herramienta para procesos de negocio.

Integración con entornos de desarrollo: apoyo al ciclo de vida completo de desarrollo de software en IDE como: Eclipse, Microsoft Visual Studio, NetBeans, Sun ONE, Oracle JDeveloper, Jbuilder y otros.

Modelado de bases de datos: generación de bases de datos y conversiones de diagramas entidad relación a tablas de bases de datos, además de mapeos de objetos y relaciones. (20)

1.6.2 Selección de la herramienta CASE

Se selecciona la herramienta Visual Paradigm para el modelado del sistema, usando UML, debido a todas sus características y facilidades de integración que ofrece, además le posibilita a los desarrolladores una plataforma con interfaz amigable que les permite diseñar un producto con calidad de forma rápida. Puede ser extendido, pues presenta soporte al diseño personalizado, lo que permite incorporar nuevas formas y

notaciones, mediante el uso de imágenes o íconos importados. Además, es la herramienta CASE más utilizada por la Universidad de las Ciencias Informáticas.

1.6.3 Herramientas para inteligencia de negocio

Las herramientas de inteligencia de negocios son un tipo de software de aplicaciones diseñadas para colaborar con la inteligencia de negocios (BI) en los procesos de las organizaciones. Específicamente se trata de herramientas que asisten el análisis y la presentación de los datos. Pese a que algunas herramientas de Inteligencia de Negocios incluyen la funcionalidad ETL (Extracción, Transformación y Carga por sus siglas en inglés), las herramientas ETL no son consideradas generalmente como herramientas de inteligencia de negocios.

1.6.3.1 Pentaho BI

Es una suite de herramientas de código abierto comercial. Las soluciones de Pentaho están desarrolladas en Java y tienen un ambiente de implementación también basado en Java. Eso hace que Pentaho sea una solución muy flexible para cubrir una amplia gama de necesidades empresariales tanto las típicas como las sofisticadas y específicas del negocio. Ofrece componentes fundamentalmente de una infraestructura de herramientas de análisis e informes integrados con un motor de *workflow* de procesos de negocio. La plataforma será capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos y actividades y de presentar y entregar la información adecuada en el momento adecuado, mediante análisis OLAP, Cuadros de Mando, etc.(21)

Componentes soportados:

Servidor: Pentaho puede correr en servidores compatibles con J2EE como JBOSS AS, IBM WebSphere, Tomcat, WebLogic y Oracle AS.
Base de datos: Vía JDBC, IBM DB2, Microsft SQL Server, MySQL, Oracle, PostgreSQL, NCR Teradata, Firebird.
Sistema operativo: No existe dependencia; lenguaje interpretado.
Lenguaje de programación: Java, Java script, JSP, XSL (XSLT / XPath / XSL-FO).
Interfaz de desarrollo: Java SWT, Eclipse, basada en la web.

Todos los componentes están expuestos vía servicios web para facilitar la integración con Arquitecturas Orientadas a Servicios (SOA).(22)

Los módulos incluidos por Pentaho BI, pueden utilizarse de manera conjunta o de forma separada según las necesidades de la organización:

- Reporte: Pentaho Reporting es una solución basada en el proyecto JFree Report y permite generar informes ágiles y de gran capacidad. Permite la distribución de los resultados del análisis en múltiples formatos. Todos los informes incluyen la opción de imprimir o exportar a formato PDF, XLS, HTML y texto. Los reportes de Pentaho permiten también programación de tareas y ejecución automática de informes con una determinada periodicidad.
- Análisis: Pentaho Analysis suministra a los usuarios un sistema avanzado de análisis de información. Con el uso de las tablas dinámicas, el usuario puede navegar por los datos, ajustando la visión de estos, los filtros de visualización, añadiendo o quitando los campos de agregación. Los datos pueden ser representados en forma de SVG (gráficos de vector escalables), Flash, dashboard widget, o también integrados con los sistemas de minería de datos y los portales web.
- Dashboard: todos los componentes del módulo Pentaho Reporting y Pentaho Analysis pueden formar parte de un Dashboard. En Pentaho Dashboard es muy fácil incorporar una gran variedad de tipos de gráficos, tablas y velocímetros e integrarlos con los portales web, en donde se podrá visualizar informes, gráficos y análisis OLAP.
- Minería de datos: Mediante Pentaho Data Mining se podrá descubrir patrones de comportamiento e indicadores ocultos en la información de una organización. Prevenir eventos futuros basados en patrones históricos para así apoyar las tareas de análisis predictivo.
- Integración de datos: se realiza con la herramienta para ETL Pentaho Data Integration (PDI, por sus siglas en inglés) que permite implementar los procesos de extracción, transformación y carga de la información.(1)

Requisitos mínimos de Pentaho BI:

- Memoria RAM: 1Gb.
- Espacio en disco duro: 1Gb.
- Procesador: Celeron 2.0 GHz.
- Necesita un JDK de java instalado con anticipación, se recomienda el JDK de Sun 1.5 o superior.

- Se necesita también los drivers JDBC de la base de datos relacional que se utilizará como fuente de datos.(1)

1.6.3.2 Spago BI

Se trata de una aplicación BI de tipo OLAP construida para acceso web y que permite acceder a datos de SQL Server y Mondrian. Es una plataforma ya que cubre y satisface todos los requisitos de BI, tanto en términos de análisis, de gestión de datos, administración y seguridad.(1)

En el mundo analítico ofrece soluciones para la presentación de informes, análisis multidimensional, minería de datos, tableros de mando y consultas ad-hoc. Añade módulos originales para la gestión de procesos de colaboración. Cuenta con herramientas para ETL y apoya al administrador en el mantenimiento de los documentos analíticos, la gestión para el control de versiones y la aprobación del flujo de trabajo. Permite generar informes perfectamente estructurados y exportarlos a multitud de formatos (HTML, PDF, XLS, XML, TXT, CSV y RTF). Además es multiplataforma y tiene licencia GNU LGPL. (1)

Estructura modular:

- Spago BI Server: núcleo central de Spago BI que integra la funcionalidad de los diferentes motores.
- Spago BI Studio: entorno de desarrollo único e integrado.
- Spago BI Meta: entorno enfocado a la capa de metadatos.
- Spago BI SDK: nivel de integración para utilizar Spago BI con aplicaciones externas.
- Spago BI Applications: para mantener los modelos verticales de análisis desarrollados con Spago BI.
- Requisitos mínimos de Spago BI:
- Memoria RAM: 512Mb.
- Servidor de aplicaciones J2EE como Tomcat, JBoss, WebSphere.(23)

1.6.4 Herramientas para el proceso de extracción, transformación y carga de datos

Las herramientas de extracción, transformación y carga de datos, proporcionan funcionalidades para obtener la información necesaria a partir de datos almacenados en fuentes externas, realizar operaciones sobre los datos para que puedan ser cargados en el AD, almacenar los datos en el AD final, control de la extracción de los datos y su automatización y proporcionar la gestión integrada del AD y los MD existentes.

1.6.4.1 Pentaho Data Integration

Desarrollado íntegramente en Java, posee licencia LGPL. Se utiliza para la integración de datos, carga de AD y MD, limpieza de datos, análisis perfilado de datos, migración de datos entre base de datos y exportar datos de bases de datos a archivos planos. Transforma e integra datos entre sistemas de información existentes y los MD que compondrán el sistema BI. Posee como principales características:

- Entorno gráfico de desarrollo.
- Uso de tecnologías estándar: Java, XML, JavaScript.
- Fácil de instalar y configurar.
- Multiplataforma: Windows, Macintosh, Linux.
- Basado en dos tipos de objetos: Transformaciones (colección de pasos en un proceso ETL) y Trabajos (colección de transformaciones).(24)

Incluye cuatro herramientas:

- Spoon: para diseñar transformaciones ETL usando un entorno gráfico.
- Pan: para ejecutar transformaciones diseñadas con Spoon.
- Chef: permite diseñar la carga de datos incluyendo un control de estado de los trabajos.
- Kitchen: permite ejecutar los trabajos batch diseñados con Chef.

Soporta diferentes fuentes de información como son: Excel, PostgreSQL, MySql, Informix, dBaseIII, IVo5, FirebirdSQL, IBMDB2, MSSQLServer, MSAccess, Oracle, SAPERP System, Teradata, LucidDB, Hypersonic y ApacheDerby.

1.6.4.2 Talend Open Studio

Tiene como principal ventaja que está implementado en Java, por lo tanto dispone de un entorno de desarrollo multiplataforma. Además, puede usar Java como lenguaje de apoyo en las tareas de transformación de datos, y se pueden crear nuevos componentes usando esté lenguaje. Todas las acciones se hacen de forma visual; Talend las transforma en código Java, que compila y entrega en forma de un archivo .jar y un script .sh o .bat; para poder ejecutarlo desde Linux, Windows o Mac. Permite también de forma visual conectar las fuentes de datos con el sistema de destino, transformando los datos mediante componentes ya creados en la aplicación.

Talend cuenta con una gran cantidad de componentes, y con una comunidad que trabaja añadiendo nuevas opciones. En cuanto a bases de datos, se pueden encontrar desde las más generales como:

MySQL, SQL Server, Oracle o PostgreSQL, o aquellas con aplicaciones más específicas como Grenplum, ParAccel o eXists. También dispone de componentes para adquirir o volcar datos utilizando ficheros de diferentes tipos: XML, Excel, delimitados (csv, tsv), JSON; e incluso la posibilidad de capturar las filas mediante expresiones regulares. (25)

1.6.5 Selección de las herramientas para BI y proceso ETL

Luego de analizar las herramientas asociadas al concepto de almacenes de datos se seleccionan Pentaho BI y dentro de este Pentaho Data Integration para el proceso ETL. Son fáciles de utilizar, gestionan la sustitución de claves y son empleadas por la comunidad de usuarios de la UCI, de la que se puede obtener un apoyo para la aplicación de las mismas.

1.6.6 Sistema gestor de base de datos

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan con un propósito general. Los sistemas de gestión de bases de datos son los encargados del manejo de manera clara, sencilla y ordenada de un conjunto de datos que posteriormente se convertirán en información relevante para una organización. (26)

1.6.6.1 MySQL

Sistema de gestión de bases de datos relacional, de propósito general y multiusuario, muy rápido, multihilo y robusto, liberado bajo una licencia dual, con una versión propietaria y una comunitaria liberada bajo la licencia GPL.

Ofrece un conjunto de funcionalidades como la portabilidad (se ejecuta en varias plataformas UNIX, Windows y MacOS X), la velocidad (usa técnicas de indexado eficiente, tablas temporales en memoria y algoritmos de optimización del JOIN), la escalabilidad (producto de su modularidad y flexibilidad en la configuración), la facilidad de uso (no se necesitan conocimientos extras para instalarlo y configurarlo), el modelo de seguridad bien distribuido (se pueden restringir los permisos de usuarios desde una base de datos completa hasta una columna) y el acceso desde otros lenguajes y sistemas (tiene librerías y APIs (del inglés *Application Programming Interface*) para la conexión a él desde Java, C/C++, Perl, PHP y TCL). (27)

Entre las características esenciales se encuentran:

- Probado con un amplio rango de compiladores diferentes
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.
- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
- Soporte a grandes bases de datos.
- Los clientes pueden conectarse al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente mysqlcheck. MySQL también incluye myisamchk, una utilidad de línea de comandos muy rápida para efectuar estas acciones en tablas MyISAM.(28)
- Al ser comprado por una compañía norteamericana, desde Cuba no se puede acceder al código fuente de la versión comunitaria que aún mantienen.(27)

1.6.6.2 PostgreSQL

Sistema de gestión de bases de datos objeto-relacional, de propósito general, multiusuario y de código abierto, liberado bajo la licencia BSD y soporta gran parte del estándar SQL. Ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión. Puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones agregadas, funciones ventanas y funciones recursivas, métodos de indexado y lenguajes procedurales.(27)

Algunas de las características más importantes y soportadas por PostgreSQL:

Generales:

- Replicación sincrónica/asincrónica.
- Copias de seguridad.
- Unicode.
- Juegos de caracteres internacionales.
- Regionalización por columna.
- Multi-Version Concurrency Control (MVCC).
- Múltiples métodos de autentificación.
- Acceso encriptado vía SSL.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

Programación y desarrollo:

- Funciones y procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL, PL/Perl, PL/Python y PL/Tcl.
- Bloques anónimos de código de procedimientos.
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, están disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML y matrices.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, etc.).

SQL:

- SQL92, SQL99, SQL2003, SQL2008.
- Llaves primarias y foráneas.
- Columnas auto-incrementales.
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash o GiST.
- Consultas recursivas.
- Funciones ventanas.
- Joins.
- Vistas.

- Disparadores comunes, por columna, condicionales.
- Reglas.
- Herencia de tablas.(29)

1.6.7 Selección del Sistema Gestor de Bases de Datos

Se decide utilizar PostgreSQL pues es un SGBD objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el SGDB de código abierto más potente del mercado, con una comunidad de desarrollo sólida, independiente y establecida; permitiendo el soporte, futuro desarrollo y continuidad. Además, al ser PostgreSQL un sistema de código abierto, adoptarlo contribuiría a incrementar la soberanía tecnológica cubana.

1.6.8 pgAdmin III

pgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usan en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones de PostgreSQL a partir de la 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.

Conclusiones del capítulo

Se decide desarrollar un almacén de datos multidimensional, empleando las definiciones OLTP y MD para la obtención y estructuración de la información respectivamente. Auxiliándose de las herramientas Visual Paradigm con UML para el modelado del sistema, Pentaho BI con Pentaho Data Integration para el proceso ETL, PostgreSQL como gestor de bases de datos con su administrador gráfico pgAdmin para la creación y gestión del almacén, y Hefesto como guía metodológica para el completo desarrollo del mismo.

Capítulo 2: Análisis, diseño, desarrollo y prueba del almacén de datos propuesto

Se realiza el análisis y diseño de la solución utilizando la metodología HEFESTO, donde una vez analizados los requerimientos se definen las preguntas de las que se identifican los indicadores y perspectivas generando el modelo conceptual. Después en el análisis de los OLTP se conforman los indicadores y se establecen las correspondencias entre el diagrama entidad relación de la plataforma Moodle y el modelo conceptual de la propuesta, permitiendo generar un modelo conceptual ampliado. Luego se realiza el modelo lógico en el que se obtuvieron las tablas dimensiones y hechos utilizando un esquema en estrella para su diseño. Se puntualiza el desarrollo del proceso ETL, en el que se apoya de la arquitectura de integración, en el que se obtienen el trabajo y las transformaciones encargadas de realizar la limpieza y carga de los datos en el AD final. Se describen las pruebas al sistema en las que se utilizaron las pruebas de integración y aceptación, que permitirán evaluar el nivel de la calidad de los datos y poder comprobar que se satisfacen las necesidades del cliente.

2.1 Análisis de requerimientos

La parte del análisis de requerimientos se lleva a cabo recolectando primeramente las necesidades de información para la herramienta de seguimiento de usuarios, obteniendo de esta manera las preguntas claves que permiten guiar la solución propuesta. Después se analizan cada una de las preguntas para identificar los indicadores y las diferentes perspectivas de análisis para luego realizar la construcción del modelo conceptual del AD. El objetivo principal de esta fase, es la de obtener e identificar las necesidades de información clave de alto nivel, que es esencial para llevar a cabo, en este caso, un modelo de usuario y la presentación de información sobre la actividad de los usuarios.

2.1.1 Identificar preguntas

Se estableció como proceso principal de análisis, la acción que realizan los usuarios dentro del EVA de la UCI, proceso que está soportado por la información registrada en la base de datos de la plataforma. Luego se procedió a identificar la información de interés acerca del proceso antes definido, descubriendo los indicadores que representan de mejor manera la actividad de los usuarios.

Para encontrar las variables o perspectivas desde las cuales se consultarán los indicadores antes enunciados se destacan las siguientes preguntas:

1. ¿Cuál es la cantidad de acciones por usuario?
2. ¿Cuál es la cantidad de acciones por usuarios en un curso?
3. ¿Cuál es la cantidad de acciones por usuario según el tipo de acción?
4. ¿Cuál es la cantidad de acciones por usuario en un módulo?
5. ¿Cuál es la cantidad de acciones por usuario desde un lugar?

Cabe recalcar que todos las preguntas anteriores pueden ser ampliadas teniendo en cuenta un periodo de tiempo (fecha, año, mes, día del mes, día de la semana, semana del año, período de clase, hora, minuto), y cualquier combinación entre las diferentes perspectivas (usuario, curso, modulo, lugar, tipo de acción, tiempo) lo cual, permitirá tener varias versiones de los datos a fin de realizar un correcto análisis posterior. La información solicitada a través de los requerimientos es la que permitirá analizar el comportamiento de los usuarios.

2.1.2 Identificar indicadores y perspectivas

Para realizar esta actividad se debe tomar en cuenta que los indicadores, para que realmente sean efectivos, son en general, valores numéricos y representan lo que se desea analizar concretamente. Por ejemplo: saldos, promedios, cantidades, sumatorias, etc.

Las perspectivas se refieren a los objetos mediante los cuales se quiere examinar los indicadores, con el fin de responder a las preguntas planteadas, por ejemplo: clientes, proveedores, sucursales, países, productos, rubros, etc. Comúnmente el tiempo es una perspectiva.

A continuación, en la Tabla 2.1 se analizan las preguntas obtenidas en el paso anterior y se detallan cuáles son sus respectivos indicadores y perspectivas.

Tabla 2.1 Identificación de Indicadores y Perspectivas.

Pregunta	Indicador	Perspectiva
Cantidad de acciones por usuario.	Cantidad de acciones.	Usuario
Cantidad de acciones por usuarios en un curso.	Cantidad de acciones.	Usuario, Curso
Cantidad de acciones por usuario según el tipo de acción.	Cantidad de acciones.	Usuario, Tipo acción
Cantidad de acciones por usuarios en un módulo.	Cantidad de acciones.	Usuario, Módulo
Cantidad de acciones por usuario desde un lugar.	Cantidad de acciones.	Usuario, Lugar

En síntesis, los indicadores encontrados en cada una de las preguntas, son:

- Cantidad de acciones.

Así mismo, las perspectivas o dimensiones encontradas en cada una de las preguntas son:

- Curso.
- Usuario (profesor, estudiante).
- Módulos.
- Tiempo (fecha, año, mes, día del mes, día de la semana, semana del año, período de clase, hora, minuto).
- Lugar.
- Tipo acción

2.1.3 Modelo conceptual

Al concluir con esta actividad se tendrá como resultado un modelo conceptual a partir de los indicadores y perspectivas obtenidas en el paso anterior. Con este modelo se podrá observar con claridad cuál es el alcance del proyecto, para luego trabajar sobre ello. Además, al poseer un alto nivel de definición de datos, permite que pueda ser representado y explicado con facilidad.

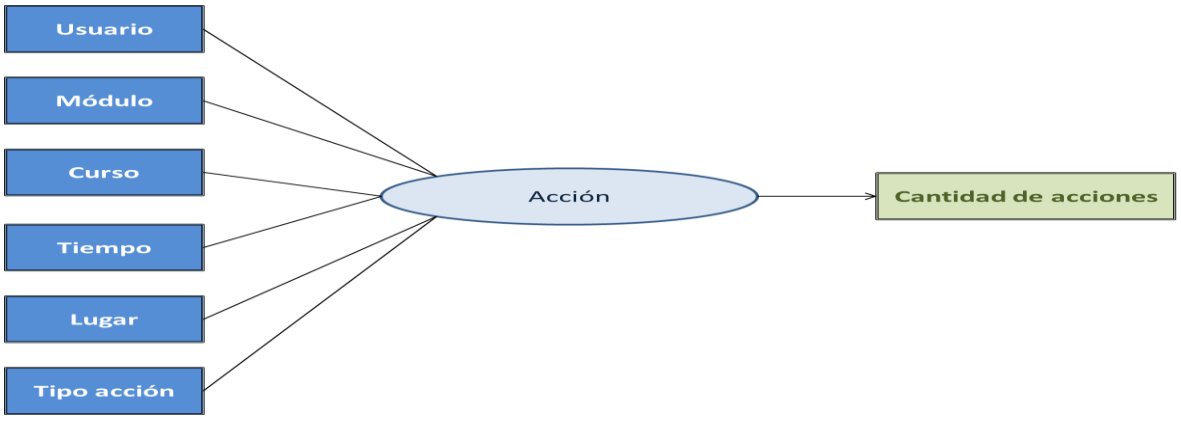


Figura 2.1 Modelo Conceptual.

A la izquierda se encuentran colocadas las perspectivas seleccionadas, que a su vez están unidas a un óvalo central que representa y lleva el nombre de la relación que existe entre ellas. La relación, constituye el proceso o área de estudio elegida. De dicha relación y entrelazadas con flechas, se desprende el indicador, el cual, se ubica a la derecha del esquema.

Como se puede observar, la relación mediante la cual se unen las diferentes perspectivas, para obtener como resultado los indicadores requeridos, es precisamente “Acción”.

2.2 Análisis de los OLTP

Se realiza el análisis de los OLTP para determinar cómo se construirán los indicadores, señalar la correspondencia con los datos fuente, seleccionar los campos de estudio de cada perspectiva y finalmente ampliar el modelo conceptual con la información obtenida en esta fase.

2.2.1 Conformar indicadores

En este paso se explica cómo se calculan los indicadores, definiendo los siguientes conceptos para cada uno de ellos:

- Indicador que lo compone, con su respectiva fórmula de cálculo.
- Función sumarización que se utilizará para su agregación.

Los indicadores se calculan tal como se puede apreciar en la Tabla 2.2:

2.2 Conformación de indicadores

Indicador	Conceptos	Aclaración
Cantidad de acciones.	-Cantidad de acciones. Función sumarización: - SUM	“Cantidad de acciones” es la sumatoria de cualquier operación registrada por el usuario en la tabla mdl_log, las cuales se pueden obtener de cualquier combinación entre las perspectivas definidas.

2.2.2 Establecer correspondencias

El objetivo de este paso, es el de examinar los OLTP disponibles que contengan la información requerida y sus características, para poder identificar las correspondencias entre el modelo conceptual y las fuentes de datos. La idea principal es, que todos los elementos del modelo conceptual estén correspondidos en los OLTP.(2)

En el OLTP de la plataforma Moodle, las visitas y acciones están representadas por el diagrama entidad relación de la Figura 2.2. Recalcar que en el diagrama solamente constan las tablas y campos principales que tienen que ver con el modelo conceptual presentado anteriormente. Se prescinde de las demás tablas

porque no permitirían una buena visualización del modelo general de la base de datos Moodle, para establecer de mejor manera las correspondencias entre los modelos y satisfacer los requerimientos.

Figura 2.2 Diagrama Entidad Relación de las principales tablas de Moodle

A continuación, se expondrá la correspondencia entre los dos modelos:

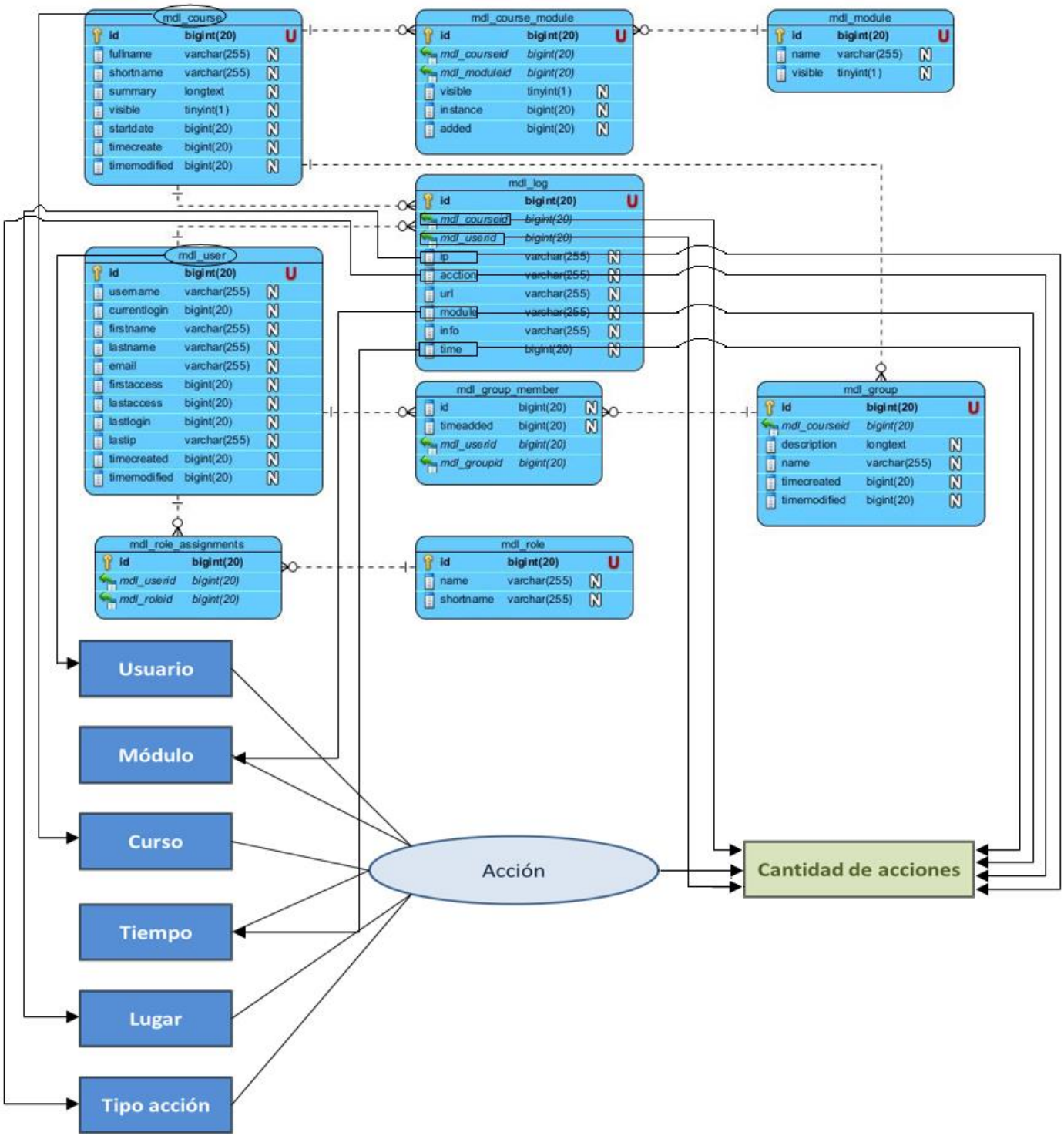


Figura 2.3 Correspondencia entre el diagrama entidad-relación de Moodle y el modelo conceptual propuesto.

Las relaciones identificadas fueron las siguientes:

- La tabla “mdl_course” se relaciona con la perspectiva “Curso”.
- La tabla “mdl_user” se relaciona con la perspectiva “Usuario”.
- El campo “module” de la tabla “mdl_log” se relaciona con la perspectiva “Módulos” porque en él se encuentran registrados todos los módulos sobre los cuales el usuario realiza sus acciones.
- El campo “time” de la tabla “mdl_log” se relaciona con la perspectiva “Tiempo” debido a que es la fecha principal para el registro de las visitas realizadas por el usuario.
- El campo “ip” de la tabla “mdl_log” se relaciona con la perspectiva “Lugar” porque en él se encuentran registradas todas las direcciones IP a través de las cuales acceden los usuarios a la plataforma, lo que permite identificar desde qué lugar se está accediendo.
- El campo “action” se relaciona con la perspectiva “Tipo acción”, ya que en este se registra el tipo de acción realizada por el usuario en el sistema.
- Los campos “time”, “mdl_courseid”, “mdl_userid”, “module” y “action” de la tabla “mdl_log” se relacionan con los indicadores “Cantidad de acciones” y “Promedio de acciones”, porque con el filtrado de los datos de la tabla “mdl_log” relacionando los campos se pueden obtener los indicadores según las necesidades del analista. Para una mejor comprensión revisar las aclaraciones definidas por cada indicador en la conformación de los mismos.

2.2.3 Nivel de granularidad

Ya que se han establecido las relaciones con los OLTP, se deben seleccionar los campos que contendrá cada perspectiva, ya que será a través de estos por los que se examinarán y filtrarán los indicadores.

Para ello, con las correspondencias establecidas, se deben presentar los datos de análisis disponibles para cada perspectiva, debido a la importancia de conocer en detalle qué significa cada campo o valor de los datos encontrados en los OLTP.

En lo que tiene que ver con la perspectiva “Tiempo”, es importante definir el ámbito mediante el cual se agruparán o sumarizarán los datos. Se determinará la granularidad de la información del AD, los campos que integrarán cada perspectiva, por lo cual, se debe prestar mucha atención en la selección de dichos campos.

De acuerdo a las correspondencias establecidas, se analizaron los campos residentes en cada tabla a la que se hacía referencia, a través de un solo método. Este método consistió en examinar la base de datos para extraer los significados de cada campo, escogiendo los más importantes para la creación del AD.

Como se puede apreciar en el diagrama entidad relación antes expuesto, algunos de los nombres de los campos son bastante explícitos pero otros no, lo cual, genera cierta confusión a la hora de determinar el significado de cada campo; esto implicó revisar también la documentación registrada en el esquema de la base de datos.

- Con respecto a la perspectiva "Curso", los datos disponibles son los siguientes:

Tabla 2.3 Datos disponibles para la perspectiva Curso.

Atributo	Descripción
id	Identificador del curso.
fullname	Nombre completo del curso.
shortname	Nombre corto (abreviatura) del curso.
summary	Descripción del curso.
startdate	Tiempo de comienzo del curso.
timecreated	Tiempo en que fue creado el curso.
timemodified	Tiempo en que fue modificado el curso.

- Con respecto a la perspectiva "Usuario", los datos que se pueden utilizar son los siguientes:

Tabla 2.4 Datos disponibles para la perspectiva Usuario.

Atributo	Descripción
id	Identificador del usuario.
username	Nombre del usuario.
firstname	Nombre de la persona que pertenece el usuario.
lastname	Apellido de la persona que pertenece el usuario.
firstaccess	Tiempo en que el usuario tuvo el primer acceso al sistema.
lastaccess	Tiempo en que el usuario tuvo el último acceso al sistema.
lastlogin	Tiempo en que el usuario tuvo el último loguin al sistema.
currentlogin	Tiempo en que el usuario tuvo el loguin actual.
lastip	Ip del último acceso del usuario.
timecreated	Tiempo en que fue creado el usuario.
timemodified	Tiempo cuando fue modificado el usuario.

- Con respecto a la perspectiva "Módulos", aunque existe una tabla "mdl_modules", los datos que se utilizan son los obtenidos del campo "module" de la tabla "mdl_log", ya que en dicha tabla se encuentran registrados los módulos del sistema y acciones que pueden ser realizadas por un usuario. A continuación la descripción de los datos:

Tabla 2.5 Datos disponibles para la perspectiva Módulo.

Atributo	Descripción
module	Nombre de los módulos.

- Con respecto a la perspectiva "Tiempo", el dato que se utilizará es el campo "time" de la tabla "mdl_log":

Tabla 2.6 Datos disponibles para la perspectiva Tiempo.

Atributo	Descripción
time	Fecha.

- Con respecto a la perspectiva "Lugar", los datos que se pueden utilizar serán obtenidos de la tabla "mdl_log", ya que en dicha tabla se encuentran registradas las diferentes IP con las cuales el usuario ha ingresado al sistema.

Tabla 2.7 Datos disponibles para la perspectiva Lugar.

Atributo	Descripción
ip	IP de conexión.

- Con respecto a la perspectiva "Tipo acción", los datos se obtendrán del campo "actión" de la tabla "mdl_log".

Tabla 2.8 Datos disponibles para la perspectiva Tipo acción.

Atributo	Descripción
action	Tipo de acción.

Una vez que se recolectó toda la información pertinente y se seleccionó cuáles eran los datos que se consideran de interés para analizar los indicadores ya expuestos, los resultados obtenidos fueron los siguientes:

Tabla 2.9 Datos seleccionados para cada perspectiva.

Perspectivas	Datos
Curso	- "id" identificador generado. - "idcourse" de la tabla "mdl_course". Ya que hace referencia al identificador de la tabla curso. - "fullname" de la table "mdl_course". Ya que hace referencia al nombre completo

	de los curso. - "shortname" de la table "mdl_course". Ya que hace referencia al nombre corto de los curso.
Usuario	- "id" identificador generado. - "iduser" de la tabla "mdl_user". Ya que hace referencia al identificador de la tabla usuario. - "username" de la tabla "mdl_user". Ya que este hace referencia al nombre del usuario. - "firstname" de la tabla "mdl_user". Ya que este hace referencia al primer apellido del usuario. - "lastname" de la tabla "mdl_user". Ya que este hace referencia al último apellido del usuario. - "namegroup" de la tabla "mdl_group". Ya que hace referencia al nombre del grupo al que pertenece el usuario.
Módulos	- "id" identificador generado. - "module" de la tabla "mdl_log". Ya que este hace referencia al módulo del sistema sobre el cual se realiza la acción.
Tiempo	- "id" identificador generado. - "time" de la acción, extraído del campo "time" de la tabla "mdl_log". - "year" ya que hace referencia al año de la acción, extraído del campo "time" de la tabla "mdl_log". - "month" ya que hace referencia al mes de la acción, extraído del campo "time" de la tabla "mdl_log". - "daymonth" ya que hace referencia al día del mes de la acción, extraído del campo "time" de la tabla "mdl_log". - "dayweek" ya que hace referencia al día de la semana de la acción, extraído del campo "time" de la tabla "mdl_log". - "namedayweek" ya que hace referencia al nombre del día de la semana de la acción, extraído del campo "time" de la tabla "mdl_log". - "dayyear" ya que hace referencia al día del año de la acción, extraído del campo

	"time" de la tabla "mdl_log". - "weekyear" ya que hace referencia a la semana del año de la acción, extraído del campo "time" de la tabla "mdl_log". - "classperiod" ya que hace referencia al periodo del año de la acción, extraído del campo "time" de la tabla "mdl_log". - "hour" ya que hace referencia a la hora de la acción, extraído del campo "time" de la tabla "mdl_log". - "minute" ya que hace referencia al minuto de la acción, extraído del campo "time" de la tabla "mdl_log". - "timelog" ya que hace referencia al atributo "time" de la tabla log sin modificación.
Lugar	- "id" identificador generado. - "ip" de la tabla "mdl_log" ya que hace referencia al IP desde el cual se conectó el usuario. - "name" que se obtiene del atributo "ip" de la tabla "mdl_log" y al comparar el nombre de las áreas asignado a cada uno de los 2do elementos de la dirección IP en el archivo xml "datosIP". Así quedaría referenciado el nombre del área desde el cual se accede al EVA.
Tipo de acción	- "id" identificador generado. - "nameaction" ya que hace referencia al nombre del tipo de acción, extraído del campo "action" de la tabla "mdl_log".

2.2.4 Modelo conceptual ampliado

En este paso, y con el fin de graficar los resultados obtenidos en los pasos anteriores, se ampliará el modelo conceptual, colocando bajo cada perspectiva los campos seleccionados y bajo cada indicador su respectiva fórmula de cálculo. Teniendo esto en cuenta se completará el diseño del diagrama conceptual como se muestra a continuación en la Figura 2.4

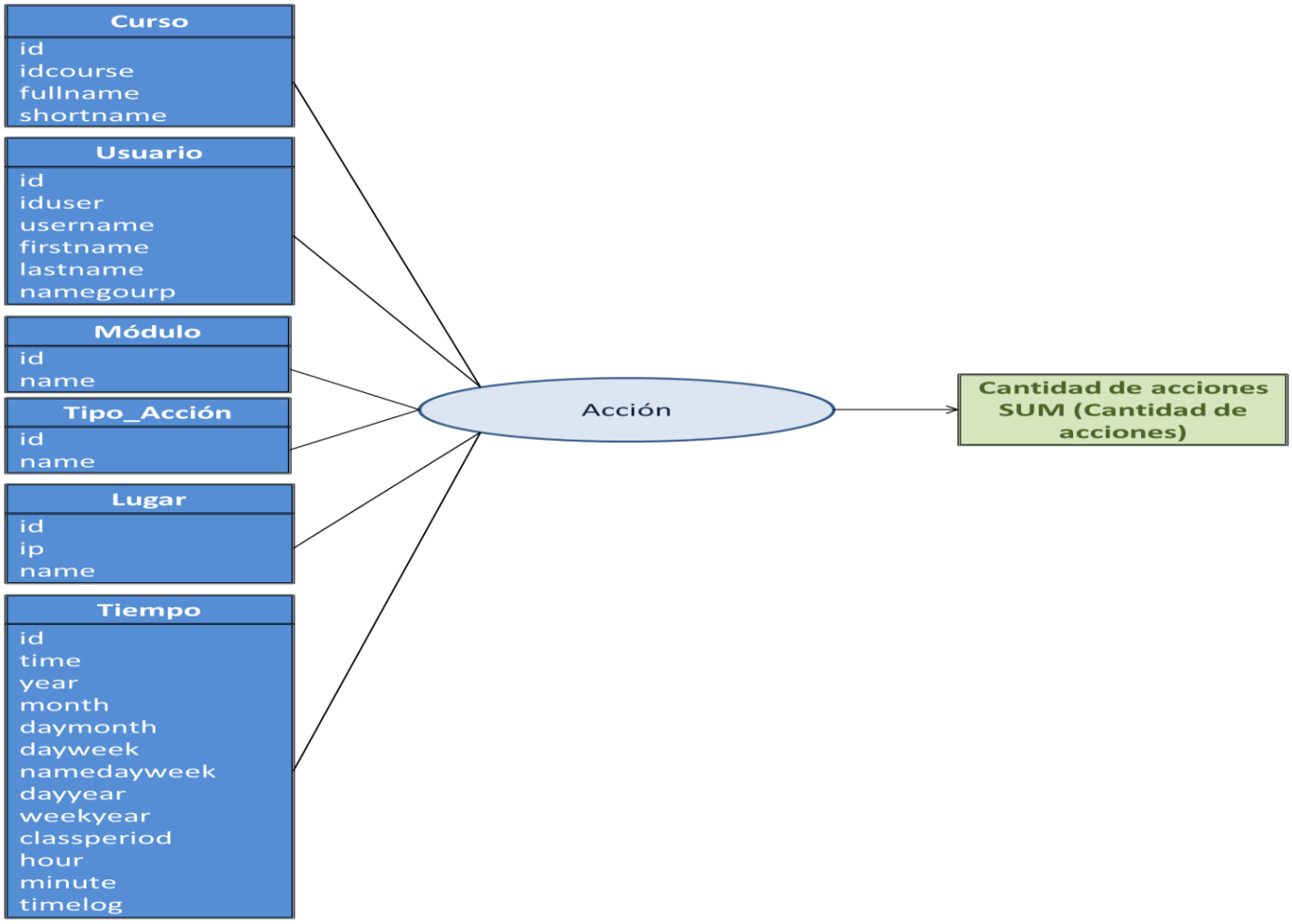


Figura 2.4 Modelo conceptual ampliado.

2.3 Modelo Lógico del AD

En este punto se definirá cuál será el tipo de esquema que se implementará para después construir las tablas de dimensiones y las tablas de hechos teniendo en cuenta los estándares de codificación definidos, y así determinar sus respectivas uniones, teniendo como base el modelo conceptual creado.

2.3.1 Tipo de Modelo Lógico del AD

Para contener la estructura del depósito de datos, de manera que se adapte a los requerimientos y necesidades ya definidos se utilizará un esquema en estrella, debido a sus características, ventajas y diferencias con otros esquemas.

2.3.2 Estándares de Codificación

Con el objetivo de organizar la estructura del almacén de datos, se formaliza un modelo, norma, patrón o estándar de codificación. Esta acción permite a los desarrolladores entender cada una de las estructuras. En la siguiente tabla se muestran como quedaron definidos estos estándares.

Tabla 2.10 Estándares de Codificación.

Estructura	Descripción	Ejemplo
Tabla de hecho	La tabla de hecho tendrá una cadena que demuestra que es un hecho y el concepto que describe.	adh_<concepto>
Tablas de dimensiones	Todas las tablas de dimensiones tendrán una cadena que demuestra que son dimensiones y el concepto que describen.	add_<concepto>
Atributos de las tablas	Todos los atributos comenzarán con la letra a con el nombre de la tabla y seguidamente el concepto que lo describa.	atabla_<concepto>
Llaves primarias	Todas las llaves primarias de cada tabla son atributos únicos y se nombrarán con el nombre de la letra a, seguido con el nombre de la tabla, luego un _ y por último con id.	atabla_id

2.3.3 Tablas de dimensiones

A continuación se muestra el diseño de las tablas de dimensiones que formarán parte del AD. Cada perspectiva definida en el modelo conceptual constituirá una tabla de dimensión. Para ello se toma cada perspectiva con sus campos relacionados y se realiza el siguiente proceso:

- Se elige un nombre que identifique la tabla dimensión.
- Se añade un campo que represente su clave principal.
- Se definen los nombres de los campos si es que no son lo suficientemente intuitivos.

Los pasos descritos anteriormente son llevados a cabo de la siguiente forma:

- Perspectiva “Curso”:

La nueva tabla de dimensión tendrá el nombre “add_curso”.

Se le agregará una clave principal con el nombre “acurso_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- **“id” por “acurso_id”.**
- **“idcourse” por “acurso_idcurso”.**

- **“fullname” por “acurso_nombre_completo”.**
- **“shortname” por “acurso_nombre_corto”.**

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

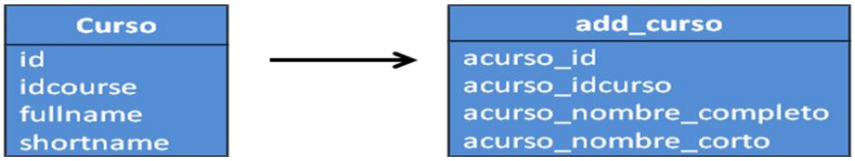


Figura 2.5 Tabla de dimensión add_curso.

- Perspectiva “Usuario”:

La nueva tabla de dimensión tendrá el nombre “add_usuario”.

Se le agregará una clave principal con el nombre “ausuario_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- **“id” por “ausuario_id”.**
- **“iduser” por “ausuario_idusuario”.**
- **“username” por “ausuario_nombre”.**
- **“firstname” por “ausuario_primer_apellido”.**
- **“lastname” por “ausuario_ultimo_apellido”.**
- **“namegroup” por “ausuario_nombre_grupo”.**

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

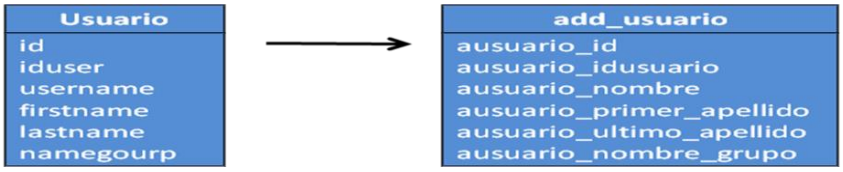


Figura 2.6 Tabla de dimensión add_usuario.

- Perspectiva “Módulo”:

La nueva tabla de dimensión tendrá el nombre “add_modulo”.

Se le agregará una clave principal con el nombre “amodulo_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- “id” de la tabla “mdl_modulo” por “amodulo_id”.
- “module” de la tabla “mdl_log” por “amodulo_nombre”.

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

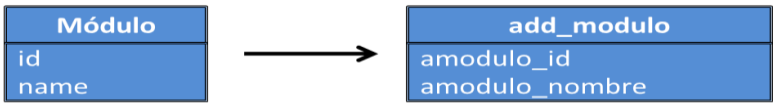


Figura 2.7 Tabla de dimensión add_modulo.

- Perspectiva “Tiempo”:

La nueva tabla de dimensión tendrá el nombre “add_tiempo”.

Se le agregará una clave principal con el nombre “atiempo_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- “year” por “atiempo_anno”.
- “month” por “atiempo_mes”.
- “daymonth” por “atiempo_dia_mes”.
- “dayweek” por “atiempo_dia_semana”.
- “namedayweek” por “atiempo_nombre_dia_semana”
- “dayyear” por “atiempo_dia_anno”.
- “weekyear” por “atiempo_semana_anno”.
- “classperiod” por “atiempo_periodo_clase”.
- “minute” por “atiempo_minuto”.
- “hour” por “atiempo_hora”.
- “timelog” por “a tiempo_tiempo_log”

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

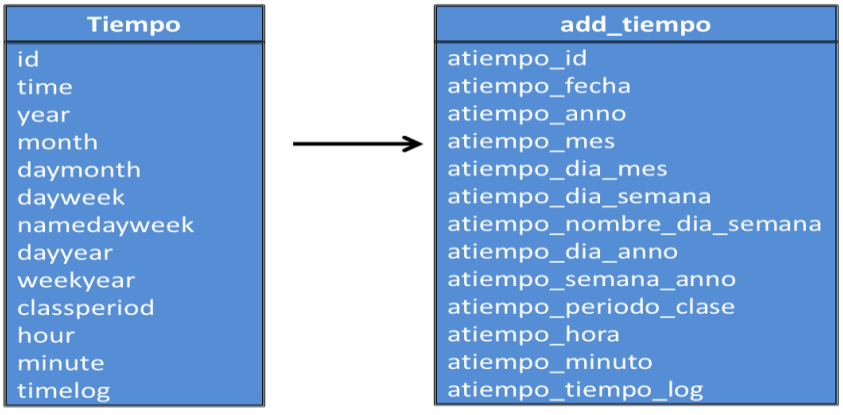


Figura 2.8 Tabla de dimensión add_tiempo.

- Perspectiva “Lugar”:

La nueva tabla de dimensión tendrá el nombre “add_lugar”.

Se le agregará una clave principal con el nombre “alugar_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- “id” por “alugar_id”.
- “ip” por “alugar_ip”.
- “name” por “alugar_nombre_lugar”.

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

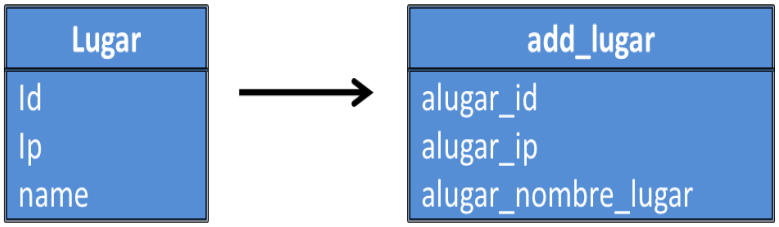


Figura 2.9 Tabla de dimensión add_lugar.

- Perspectiva “Tipo de acción”:

La nueva tabla de dimensión tendrá el nombre “add_tipo_accion”.

Se le agregará una clave principal con el nombre “atipoaccion_id”.

La modificación de los nombres de campo se realiza de la siguiente manera:

- “id” por “atipoaccion_id”.
- “name” por “atipoaccion_lugar”.

Se puede apreciar el resultado de estas acciones en la siguiente gráfica:

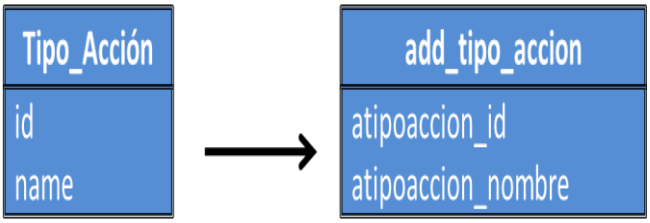


Figura 2.10 Tabla de dimensión add_tipo_accion**.**

2.3.4 Tabla hecho

La tabla hecho que se definirá a continuación, contendrá los hechos a través de los cuales se construirán los indicadores de estudio. La definición de esta tabla se realizará bajo los siguientes criterios:

- Se asigna un nombre a la tabla hecho que represente la información analizada.
- Se define la clave primaria, que se compone de la combinación de las claves primarias de cada tabla de dimensión relacionada.
- Se crean tantos campos de atributos como indicadores se hayan definido en el modelo conceptual y se les asignan los mismos nombres que estos, o cualquier otro nombre si así se prefiere.

Tomando en cuenta los criterios ya descritos la confección de la tabla hecho sería de la siguiente forma:

- La tabla hecho tendrá el nombre “adh_accion”.
- Su clave principal será la combinación de las claves principales de las tablas de las dimensiones antes definidas: “acurso_id”, “ausuario_id”, “amodulo_id”, “atiempo_id”, “alugar_id”, “atipoaccion_id”.
- Se creará 1 medidas, que se corresponden con el indicador antes definido y será nombrado, “Cantidad de acciones” por “amcantidad_acciones”.

Para apreciar de mejor manera el diseño de la tabla de hechos se puede observar la siguiente figura:

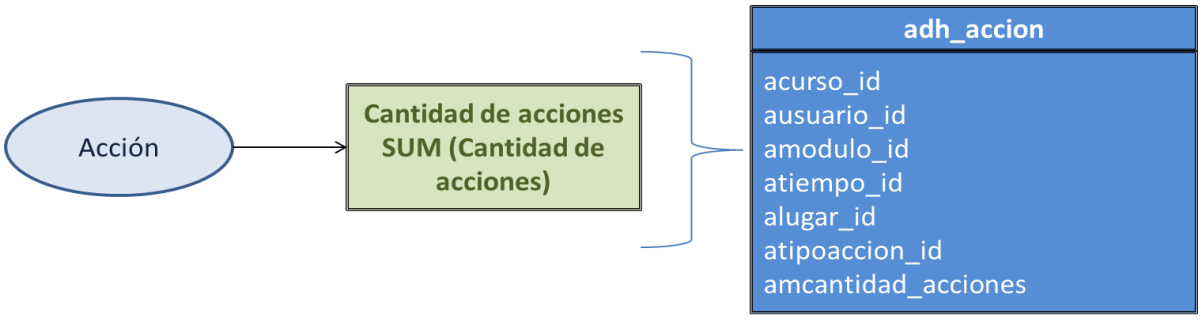


Figura 2.11 Tabla hecho adh_accion.

2.3.5 Uniones

A continuación se realizarán las uniones correspondientes entre las tablas de dimensión y la tabla hecho. En la siguiente figura se muestra el resultado de esta actividad:

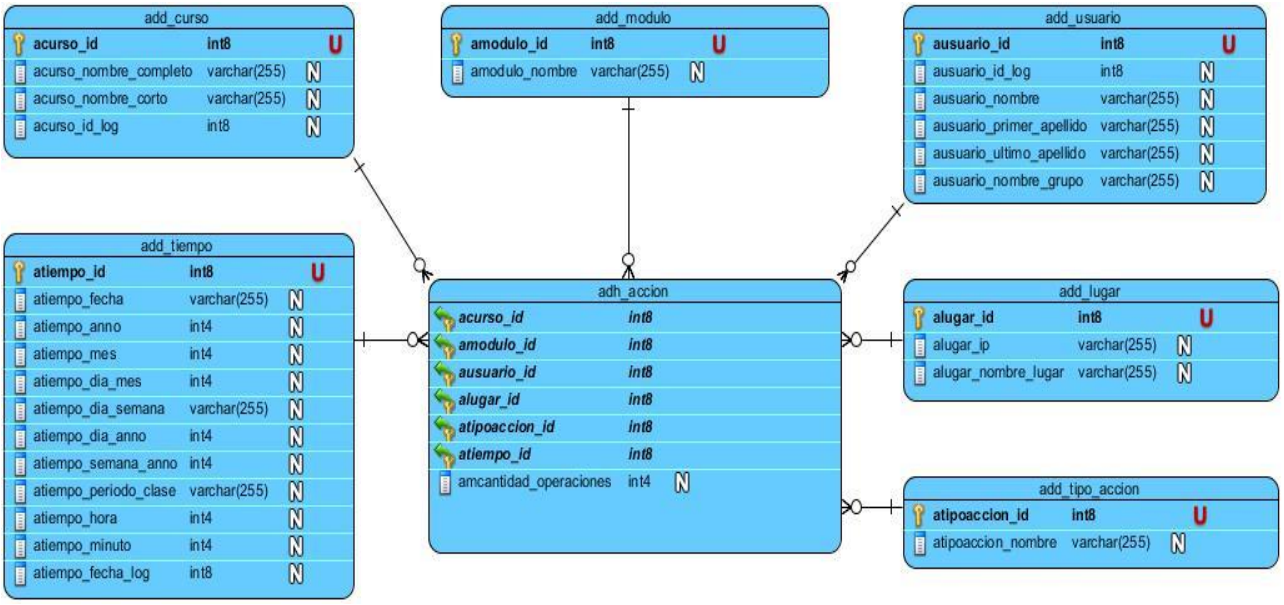


Figura 2.12 Modelo Lógico del AD.

2.4 Integración de datos

Una vez construido el modelo de datos del AD, se deberá proceder a poblarlo con datos. Se definirán políticas y estrategias para la carga inicial del AD y su respectiva actualización utilizando técnicas de limpieza y calidad de datos y procesos ETL.

2.4.1 Arquitectura de integración

En el proceso de desarrollo de un software, la arquitectura es el diseño más importante en la estructura del sistema, debido a que permite guiar la construcción del mismo. Para un correcto desarrollo del almacén es recomendable que se analice rigurosamente todo el proceso de integración de los datos. En este proceso la arquitectura está fundamentada por algunos elementos necesarios para la correcta implementación del sistema lo que se describen a continuación:

- La base de datos del EVA que representa la fuente de datos montada en MySQL.
- El archivo XML datosIP que contendrá los segundos elementos de los IP de la UCI con su respectivo lugar.
- Proceso ETL; es el punto intermedio entre la fuente y el almacén, es donde se realiza la extracción, transformación y carga de los datos, utilizando la herramienta Pentaho Data Integration de Pentaho BI.

- El almacén de datos el cual constituye el destino donde se cargan los datos, montado en PostgreSQL utilizando como cliente gráfico pgAdmin III.
- Como protocolo de comunicación TCP/IP.

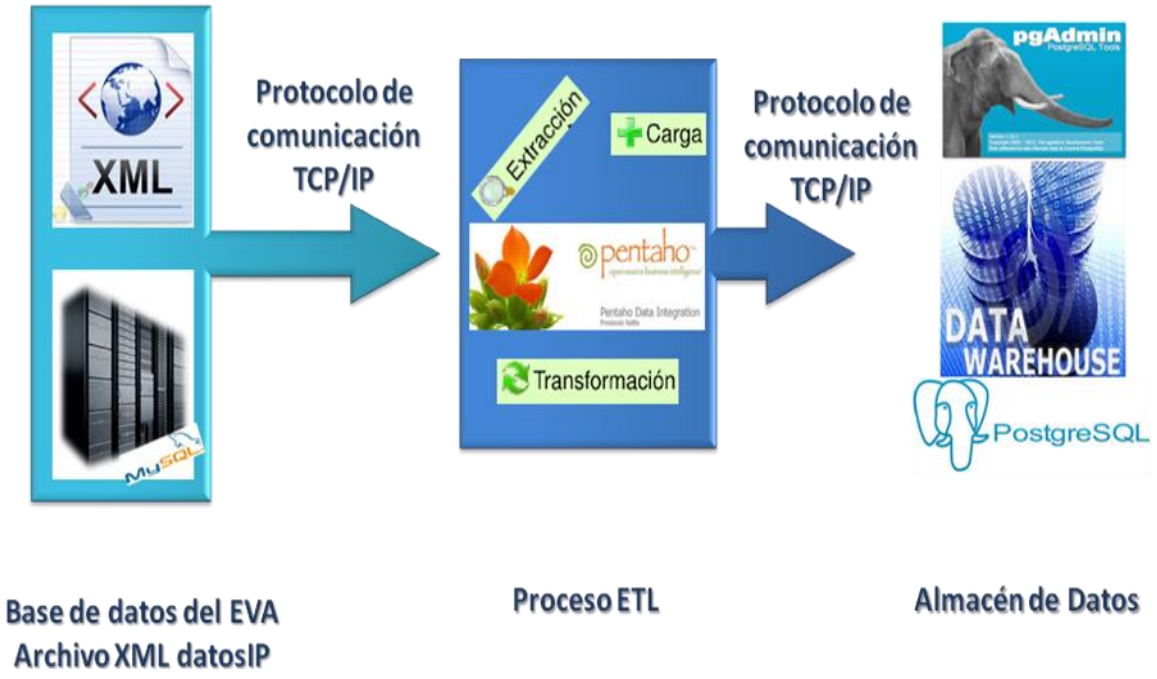


Figura 2.13 Arquitectura de Integración.

2.4.2 Implementación del proceso ETL

Los procesos de ETL en la herramienta PDI se efectúan a través de transformaciones y trabajos. A continuación se describen estos procesos:

- **Extracción:**

El primer proceso de ETL consiste en extraer los datos desde los sistemas de origen, que pueden ser provenientes de diferentes fuentes. A través de las conexiones a las fuentes se establece desde dónde se extraerán los datos para analizarlos. En este caso particular los datos son extraídos de la base de datos del EVA. La extracción se realizó a través del componente Entrada de Tabla.

- **Transformación:**

La transformación es el proceso básico de ETL, se compone de pasos que están enlazados a través de saltos. Los pasos son los elementos más pequeños dentro de las transformaciones. Los saltos son el medio por donde fluye la información entre los diferentes pasos. Después de realizada la extracción de los

datos el sistema se encuentra listo para la etapa de transformación. Durante el proceso se llevaron a cabo tareas tales como: unión por clave, validación de campos nulos; así como asignación de llaves para relacionar la información de los hechos con las dimensiones.

- **Carga:**

La carga es el último subproceso dentro de los procesos de ETL, el cual consiste en cargar todos los datos que ya han sido transformados satisfactoriamente en el almacén de datos. La carga de los datos se realizó a través de la opción Insertar/Actualizar.

2.4.3 Implementación de las Transformaciones

Las trasformaciones realizadas a los datos obtenidos de la base de datos del EVA fueron las siguientes:
Para las transformaciones de las dimensiones, “modulo”, “curso”, “tipo_accion” se utilizaron dos pasos:

- Entrada tabla.
- Insertar/Actualizar.

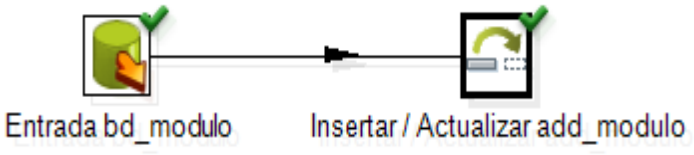


Figura 2.14 Transformación de la dimensión Módulo.

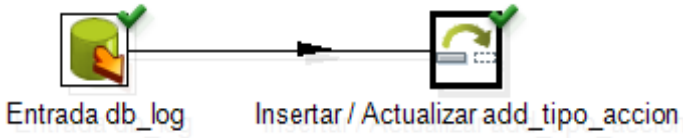


Figura 2.15 Transformación de la dimensión Tipo Acción.

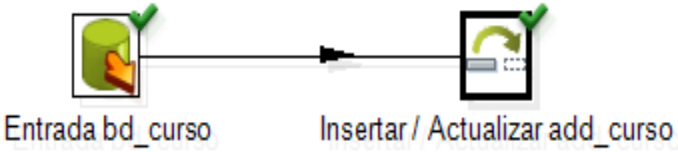


Figura 2.16 Transformación de la dimensión Curso.

Para la transformación de la dimensión “usuario” se utilizaron los siguientes pasos:

- Entrada tabla.
- Búsqueda en Flujo de Datos.
- Insertar/Actualizar.

Figura 2.17 Transformación de la dimensión Usuario.

Para la transformación de la dimensión "lugar" se utilizaron los siguientes pasos:

- Entrada tabla.
- Script.
- Get data from XML.
- Búsqueda en Flujo de Datos.
- Insertar/Actualizar.

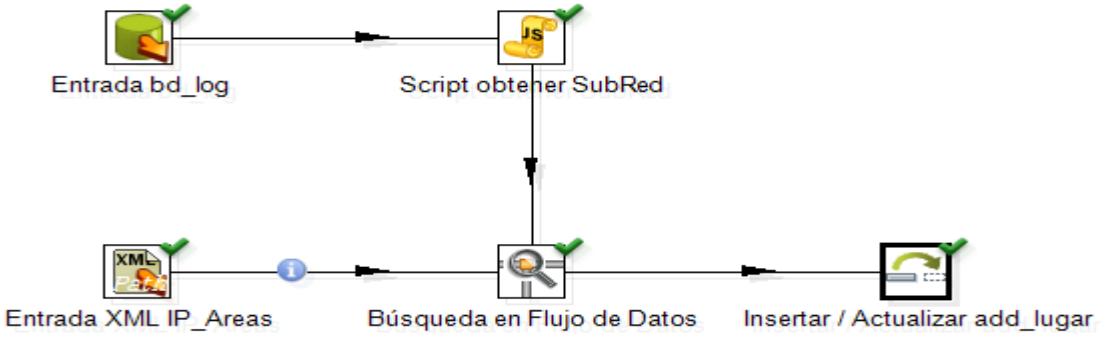


Figura 2.18 Transformación de la dimensión Lugar.

Para la transformación de la dimensión "fecha" se utilizaron los siguientes pasos:

- Entrada tabla.
- Calculadora.
- Number range.
- Replace in string.
- Script.
- Insertar/Actualizar.

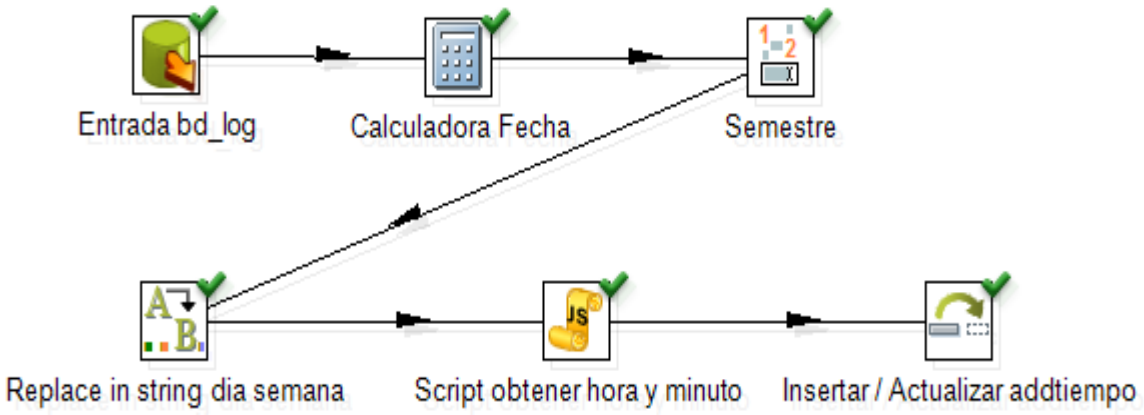


Figura 2.19 Transformación de la dimensión Fecha.

Para la transformación del hecho "accion" se utilizaron los siguientes pasos:

- Entrada tabla.
- Búsqueda en Base de Datos
- Insertar/Actualizar.

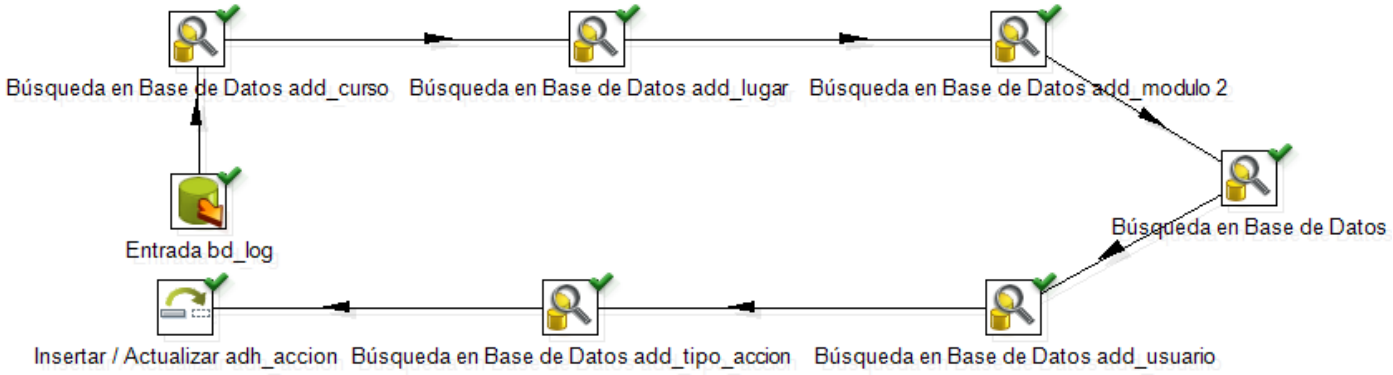


Figura 2.20 Transformación del hecho Acción.

2.4.4 Implementación de los trabajos

Un trabajo es un conjunto de tareas con el objetivo de realizar una acción determinada. En los trabajos se utilizan pasos específicos que son diferentes a los disponibles en las transformaciones. Permite ejecutar una o varias transformaciones de las diseñadas siguiendo una secuencia de ejecución. Después que se realizaron las transformaciones a los datos se organizó la carga de las tablas al almacén de datos. A continuación será descrito este proceso:

En la propuesta se realizó un trabajo que chequearía las conexiones, luego la ejecución de las transformaciones que a su vez cargarían los datos en el almacén de datos final. En la figura 2.21 se muestran los pasos realizados en este trabajo.

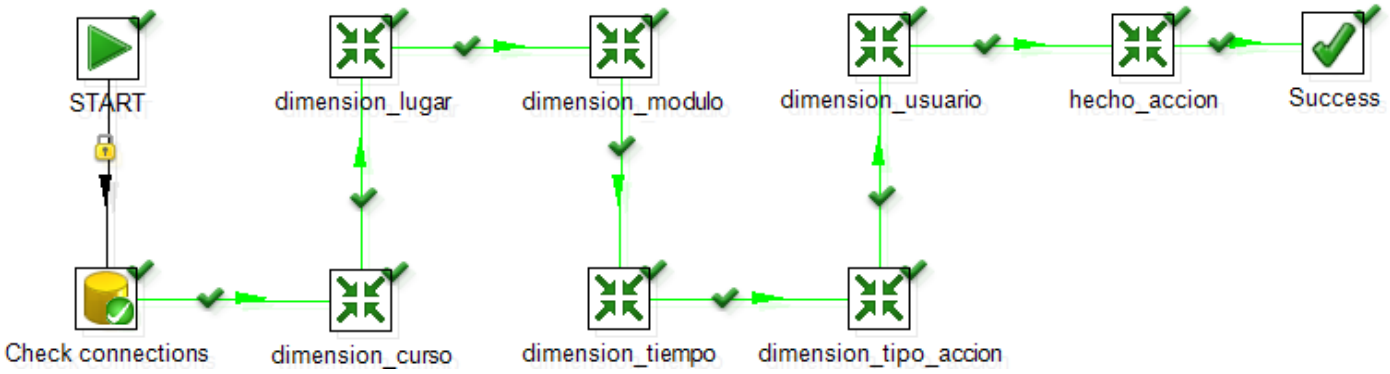


Figura 2.21 Trabajo para chequear las conexiones y ejecutar las transformaciones.

2.5 Guía de implantación

La guía de implantación contiene los pasos necesarios para la implantación de cualquier sistema informático. Antes de abordar los pasos necesarios para lograr esta implantación, es preciso conocer los requisitos del sistema los cuales constituyen requisitos no funcionales que este debe cumplir.

Requisitos del sistema:

- 500 GB de capacidad de almacenamiento.
- Red a 100 Mbps o más.

Pasos de implantación de la solución:

- Instalar máquina virtual de java (java-6-openjdk o superior).
- Instalar el SGBD PostgreSQL 9.1.
- Se debe crear una base de datos siguiendo el modelo lógico antes diseñado.
- Crear una tarea programada que ejecute el programa Kitchen de PDI para realizar la carga de los datos utilizando el trabajo desarrollado. Ver anexo 1.

2.6 Prueba

El desarrollo de un software involucra una serie de acciones de producción en las que existen altas posibilidades de que se cometan errores a la hora de su implementación. Se pueden manifestar desde el primer momento en que los objetivos o requerimientos especificados estén de forma errónea e imperfecta, de la misma forma en los pasos del diseño y desarrollo. Producto al interés humano de trabajar y comunicarse de forma perfecta, la elaboración o desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta el momento. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Existen diferentes tipos de pruebas, cada uno aplicable en un entorno diferente de acuerdo a los objetivos que se persigan en su realización. Para verificar el correcto funcionamiento del almacén de datos se realizaron las siguientes pruebas.

2.6.1 Pruebas de integración

Las pruebas de integración tienen como objetivo identificar errores introducidos por la combinación de programas o componentes probados unitariamente, permitiendo validar el flujo de control entre los módulos, y sobre los datos que son intercambiados entre ellos.

Para comprobar la correcta integración de los subsistemas de almacenamiento y ETL, se diseñaron siete casos de prueba, uno por cada transformación. Estos casos de pruebas se especifican de acuerdo a cada uno de las transformaciones que se diseñaron para la carga de las dimensiones y del hecho de la solución, y están enfocados a verificar la adecuada correspondencia entre el flujo de comunicación y el paso de atributos de la fuente de origen al destino.

2.6.1.1 Casos de pruebas

Los casos de prueba diseñados cuentan con cinco columnas en las cuales se recogen los siguientes parámetros: escenario, nombre de la transformación, extracción de la transformación, visualización de la transformación, carga de la transformación.

A continuación se muestran los casos de pruebas diseñados para la solución:

Tabla 2.11 Caso de prueba para la transformación de la dimensión "Usuario".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC1	"dimensión_usuario"	SELECT DISTINCT u.`firstname`, u.`lastname`, u.`username`, u.`id` FROM `mdl_log` as l inner join `mdl_user` as u on l.`userid`= u.`id` SELECT Distinct g.`name`,u.`id` FROM `mdl_user` as u inner join `mdl_groups_members` as gm on u.`id`=gm.`userid` inner join `mdl_groups` as g on gm.`groupid`=g.`id`	Debe visualizar del usuario, el nombre y los apellidos, el id de la tabla "mdl_user" y el grupo al que pertenece.	Se debe cargar del usuario, el nombre y los apellidos, el id de la tabla "mdl_user", el grupo al que pertenece y autogenerar un id que sería la llave primaria en el almacén.

Tabla 2.12 Caso de prueba para la transformación de la dimensión "Curso".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC2	"dimensión_curso"	SELECT c.`id`, c.`fullname`, c.`shortname` FROM `mdl_course` as c JOIN `mdl_log` as l on l.`course`= c.`id`	Debe visualizar del curso, el nombre completo y el corto y el id de la tabla "mdl_course".	Se debe cargar del curso, el nombre completo y el corto, el id de la tabla "mdl_course" y autogenerar un id que sería la llave primaria en el almacén.

Tabla 2.13 Caso de prueba para la transformación de la dimensión "Modulo".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC3	"dimensión_modulo"	SELECT l.`module` FROM `mdl_log` as l	Debe visualizar del módulo el nombre.	Se debe cargar del módulo, el nombre y autogenerar un id que sería la llave primaria en el almacén.

Tabla 2.14 Caso de prueba para la transformación del hecho "Accion".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC7	"hecho_accion"	SELECT l.`action`, l.`course`, l.`ip`, l.`module`, l.`time`, l.`userid` from `mdl_log` as l inner join `mdl_user` as u on l.`userid`= u.`id` inner join `mdl_groups_members` as gm on u.`id`=gm.`userid` inner join `mdl_groups` as g on gm.`groupid`=g.`id`	Debe visualizar de la tabla "mdl_log", el id del usuario y del curso, el tiempo, ip, nombre del módulo y la acción.	Se debe cargar en el hecho acción, el id del usuario de la tabla "adusuario", id del curso de la tabla "adcurso", id del tiempo de la tabla "adtiempo", id del ip de la tabla "adlugar", id del módulo de la tabla "admodulo" y el id de la acción de la tabla "adtipo_accion" y el id autogenerado que sería la llave primaria en el almacén.

Tabla 2.15 Caso de prueba para la transformación de la dimensión "Tiempo".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC6	"dimensión_tiempo"	SELECT DISTINCT FROM_UNIXTIME(`mdl_log`.`time`) as tiempo, `mdl_log`.`time` as tiempo_log from `mdl_log`	Debe visualizar del tiempo, la fecha en formato unixtime y la fecha en formato bigint.	Se debe cargar del tiempo, el id autogenerado que sería la llave primaria, fecha, año, mes, día del mes, día de la semana, nombre del día de la semana, día del año, semana del año, período de clase, hora, minuto y la fecha en tiempo bigint.

Tabla 2.16 Caso de prueba para la transformación de la dimensión "Lugar".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC4	"dimensión_lugar"	SELECT DISTINCT ip FROM `mdl_log` Acceder al fichero datosIP.xml para obtener los datos sobre el lugar al que pertenezcan las direcciones IP.	Debe visualizar del lugar, el ip, del fichero xml la subred y el nombre del área.	Se debe cargar del lugar, ip, el nombre del área al que pertenece, el ip y autogenerar un id que sería la llave primaria en el almacén.

Tabla 2.17 Caso de prueba para la transformación de la dimensión "Tipo accion".

Escenario	Nombre Transformación	Extracción Transformación	Visualización Transformación	Carga Transformación
ESC5	"dimensión_tipo_accion"	SELECT l.`action` FROM `mdl_log` as l	Debe visualizar del tipo acción, el nombre.	Se debe cargar del tipo acción, el nombre y autogenerar un id que sería la llave primaria en el almacén.

2.6.1.2 Resultado de las pruebas de integración

Aplicadas las pruebas de integración usando los casos de pruebas en una primera interacción se detectaron cinco No Conformidades (NC), de ellas tres tienen complejidad media y dos de complejidad alta, las NC con su descripción se especifican a continuación:

NC1: En el ESC1 al cargar la transformación en el almacén no se guardó el grupo al que pertenecían los usuarios.

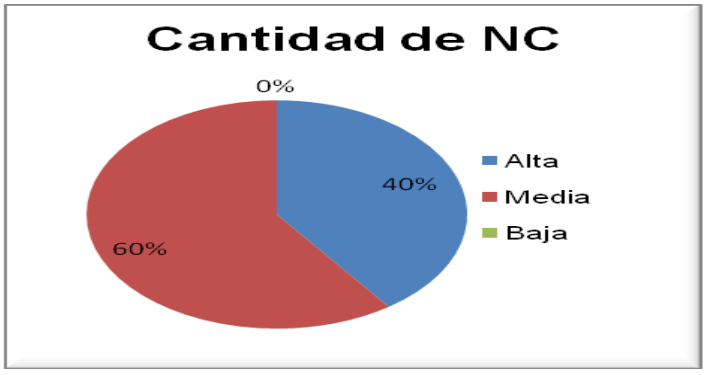
NC2: En el ESC4 al cargar la transformación en el almacén no se guardó el nombre del área.

NC3: En el ESC 6 al cargar la transformación en el almacén no se guardó la hora ni el minuto.

NC4: En el ESC 7 al ejecutar la transformación no se encontró el atributo "ausuario_id_log" del usuario de la tabla "adusuario".

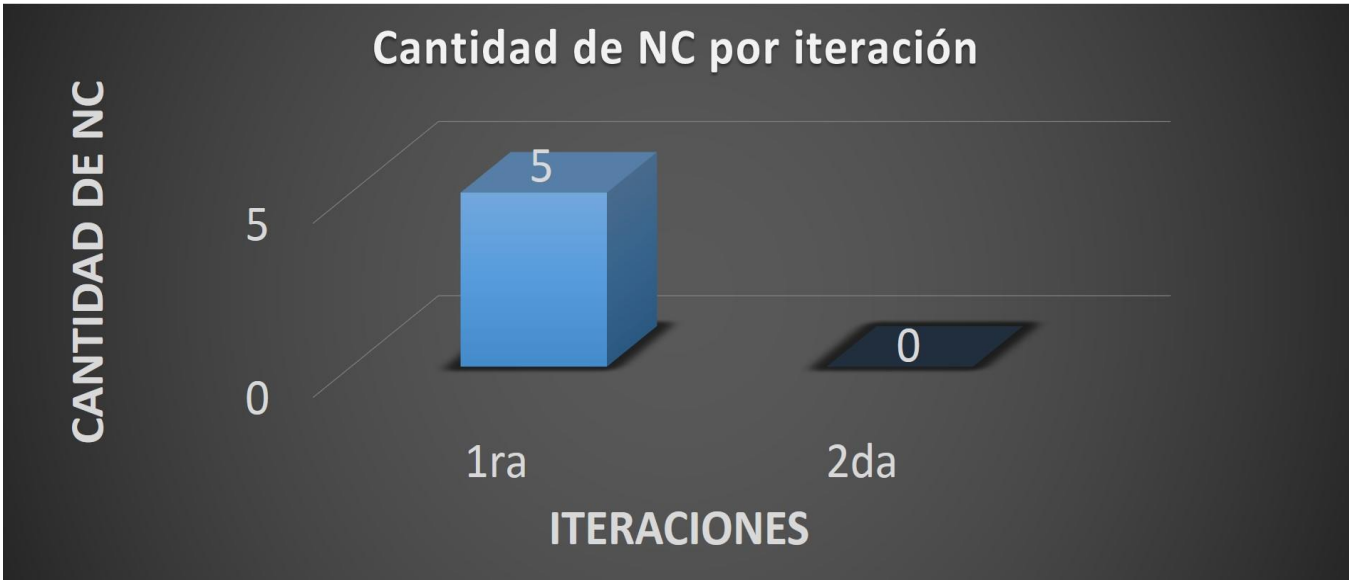
NC5: En el ESC 7 al ejecutar la transformación no se encontró el atributo "acurso_id_log" del curso de la tabla "adcurso".

En la siguiente gráfica se desglosan las NC según el impacto.



Gráfica 2.1 Especificación de las NC contra el imapcto.

Después de detectadas las NC en la primera iteración se procedió a su resolución y luego de realizarse la segunda iteración no se detectaron NC. A continuación se muestra un gráfico que representa la cantidad de NC por iteración:



Gráfica 2.2 Especificación de las NC por interación.

2.6.2 Pruebas de aceptación

Las pruebas de aceptación tienen como objetivo validar que un sistema cumple con el funcionamiento que se espera y permitir que el usuario de este determine su aceptación, desde el punto de vista del funcionamiento y rendimiento de dicho sistema.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, en este caso por el autor del presente trabajo, aunque la ejecución y aprobación final corresponden al usuario. Para validar la solución se realizó un encuentro con el cliente, donde el mismo realizó las pruebas de validación que demostraron la conformidad con la aplicación. Ver anexo 2.

Conclusiones del capítulo

Del proceso de análisis y diseño del AD se identificaron 5 preguntas, de las que se determinaron las perspectivas, indicador y el hecho que relaciona dichos conceptos. A partir de este proceso se diseñaron diferentes modelos, se establecieron las correspondencias entre el diagrama entidad-relación de Moodle y el modelo conceptual, dando paso al modelo conceptual ampliado y este a su vez al modelo lógico de los

datos correspondiente al almacén. En el proceso de ETL se realizó un trabajo con siete transformaciones, tomando como base la arquitectura de integración entre el almacén desarrollado, la base de datos del EVA y el archivo XML datos IP. Se describe una guía de pasos para la implantación del almacén de datos obtenido. Las pruebas de integración fueron desarrolladas en cuanto a las transformaciones de las dimensiones y del hecho, diseñándose 7 casos de pruebas, en los que en una primera interacción se señalaron 5 NC, las cuales fueron corregidas. Una vez ejecutada la segunda iteración no se detectaron ninguna otra NC. Según esto y la validación de los datos integrados con las pruebas de aceptación se garantiza la calidad de los mismos y el cumplimiento a las necesidades del cliente.

Conclusiones generales

Luego de desarrollar un AD para estructurar la información generada por la interacción de los usuarios con el EVA, se pudo arribar a las siguientes conclusiones:

- El estudio de las herramientas, metodologías, sistemas existentes y conceptos comunes en el desarrollo de AD, permitió establecer el perfil tecnológico.
- Mediante el análisis y diseño de la propuesta se obtuvo como resultado un modelo de datos relacional que representa las 6 dimensiones y el hecho de la propuesta.
- Con los procesos de extracción, transformación y carga de datos, se obtuvo un AD, que constituye una solución viable para almacenar y estructurar adecuadamente la información generada por la interacción de los usuarios con el EVA lo que facilita su posterior análisis.
- Los resultados satisfactorios de las pruebas de integración y la validación de los datos integrados con las pruebas de aceptación garantizan la calidad del almacén y por consiguiente el cumplimiento a las necesidades del cliente.

Recomendaciones

Luego de realizada la investigación y teniendo en cuenta las ideas surgidas durante su desarrollo, se recomienda:

- En la creación de herramientas de integración de información como lo es esta propuesta, se deben analizar los requerimientos de hardware, ya que estos sistemas comprometen el rendimiento total de la computadora o servidor sobre el cual se ejecuta.
- Integrar la visualización de los reportes a la plataforma educativa EVA.
- Mantener actualizada la información del almacén de datos, incorporando progresivamente una mayor cantidad de datos para que se realice un análisis más detallado de los mismos.

Bibliografías

1. VITIER URQUIZU. Almacén de datos operacional para contribuir a la toma de decisiones basado en el uso de los recursos de la plataforma educativa ZERA. Habana, Cuba : Universidad de la Ciencias Informáticas, 2013.

2. MALDONADO NARVÁEZ. SISTEMAS ADAPTATIVOS EN MOODLE: MÓDULO DE SEGUIMIENTO DE USUARIOS. Trabajo de fin de carrera previo a la obtención del título de Ingeniero en Sistemas Informáticos y Computación. Loja, Ecuador : Universidad Técnica Particular de Loja, 2010.

3. FUSION.DESIRE2LEARN. Sesiones. [online]. [Accessed 2 April 2014]. Available from: http://fusion.desire2learn.com/es/sesiones/

4. DURÁN QUINTANA. Modelo basado en una comunicación en tiempo real para plataformas e-Learning en la web. [online]. 2012. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_06186_12

5. LAIME PAISANT, GONZÁLEZ RICARDO and HIDALGO GUILLÉN. Extensión del módulo Chatxmpp de la plataforma de teleformación Moodle [online]. 2012. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_05565_12

6. MOODLE. Acerca de Moodle - MoodleDocs. [online]. 2014. [Accessed 10 March 2014]. Available from: http://docs.moodle.org/all/es/Acerca_de_Moodle

7. RALPH KIMBALL. The Data Warehouse ETL Toolkit. 2002.

8. CESARES, C. Data Warehousing. México : Instituto Tecnológico de Chihuahua., 2011.

9. INMON, B. Building the Data Warehouse. s.l. : Wiley,, 1992.

10. BERNABEU RICARDO. DATA WAREHOUSING: Investigación y Sistematización de Conceptos - HEFESTO: Metodología propia para la Construcción de un Data Warehouse. Córdoba,Argentina : s.n., 2010.

11. DATAPRIX. Data warehouse manager. Metodología datawarehousing | Manual IT online. [online]. 2014. [Accessed 3 March 2014]. Available from: http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/arquitectura-del-data-warehouse/34-datawarehouse-manager

12. NUEVA AGUILAR. Construcción de un Data Mart para el Sistema Integral de Gestión de Medicamentos. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Habana, Cuba : Universidad de las Ciencias Informáticas, 2011.

13. BERNABEU RICARDO, Dario. Hefesto. Córdoba, Argentina, 2009.

14. MARTA ZORRILLA. Data warehouse y OLAP. 2009. Universidad de Cantabria.

15. DATAPRIX. Herramientas ETL. ¿Que son, para que valen?. Productos mas conocidos. ETL´s Open Source. | Integración de datos. [online]. 2014. [Accessed 8 March 2014]. Available from: http://www.dataprix.com/blogs/respinosamilla/herramientas-etl-que-son-para-que-valen-productos-mas-conocidos-etl-s-open-sour

16. BRETONES LORENZO and AGUILAR FERNÁNDEZ. Procedimiento para la gestión documental para productos de tipo de almacenes de datos. La Habana : Universidad de las Ciencias Informáticas, 2011.

17. CARRILLO A. Herramienta Multimedia de apoyo a la Enseñanza de la Metodología RUP de Ingeniería del Software. Edición electrónica gratuita. 2009.

18. DE LA CRUZ RODRIGUEZ, GUZMÁN HERNÁNDEZ and PÉREZ JAVIER. Propuesta de herramienta CASE para los proyectos del Centro de Desarrollo de Informática Industrial (CEDIN) [online]. 2010. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_03818_10

19. CABRERA GONZÁLEZ and POMPA TORRES. Extensión de Visual Paradigm for UML para el Desarrollo Dirigido por Modelos de aplicaciones de gestión de información. [online]. 2013. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/JCE-2012-F323-P164-Ponencia-2777

20. LEDESMA RODRÍGUEZ, LÓPEZ DUQUE, BOZA ROGET and ROBERT LOBO. Extensión de la herramienta Visual Paradigm for UML para el soporte al Desarrollo Dirigido por Modelos con Ext JS. [online]. 2011. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_04358_11

21. DATAPRIX. Pentaho Business Intelligence (BI) | Manual IT online. [online]. 2014. [Accessed 4 March 2014]. Available from: http://www.dataprix.com/72-pentaho-business-intelligence-bi

22. DATAPRIX. Componentes del Pentaho | Manual IT online. [online]. 2014. [Accessed 4 March 2014]. Available from: http://www.dataprix.com/722-componentes-pentaho

23. BEYENETWORK. SpagoBI, una suite Business Intelligence completamente Open Source by Josep Curto Díaz - BeyeNETWORK Edición Español. [online]. 2014. [Accessed 4 March 2014]. Available from: http://www.beyenetwork.es/view/10428

24. SLIDESHARE. Pentaho PDI. [online]. 2014. [Accessed 4 March 2014]. Available from: http://es.slideshare.net/mpierri/pentaho-pdi

25. BRUJULEO. Introducción a Talend - Brujuleo. Brujuleo [online]. 2014. [Accessed 4 March 2014]. Available from: http://www.brujuleo.es/introduccion-a-talend/

26. DAMARYS QUEVEDO and GONZÁLEZ GUERRA. Sistema para la gestión de la información de postgrado en la facultad 3 [online]. 2012. [Accessed 3 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_05479_12

27. VAZQUEZ ORTÍZ and PIÑERO PÉREZ. Estrategia para la obtención de un gestor de bases de datos cubano. [online]. 2013. [Accessed 8 March 2014]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/7996

28. MYSQL. MySQL :: MySQL 5.0 Reference Manual :: 1.4.2 Las principales características de MySQL. [online]. 2014. [Accessed 8 March 2014]. Available from: http://dev.mysql.com/doc/refman/5.0/es/features.html

29. POSTGRESQL. Sobre PostgreSQL | www.postgresql.org.es. [online]. 2014. [Accessed 3 March 2014]. Available from: http://www.postgresql.org.es/sobre_postgresql

Anexos

Anexo 1: Actualización del AD

Para actualizar la base de datos se debe crear una tarea que se ejecute a las 3:00 am de todos los días, que a su vez implemente la siguiente línea:

D:\Tesis\Tesis\Herramientas\Pentaho\design-tools\data-integration\Kitchen.bat
/file:"..D:\Tesis\Tesis\Almacen\almacen tesis\job\Hecho_Accion.kjb "

Aclarar que se debe tener en cuenta las direcciones en donde se encuentran los ficheros Kitchen.bat y Hecho_Accion.kjb para que sustituyan las direcciones.

Anexo 2: Opinión del cliente

El Trabajo de Diploma, titulado Almacén de datos para el Entorno Virtual de Aprendizaje, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

El presente trabajo es una base para apoyar la toma de decisiones de los profesores con respecto a las actividades montadas en el entorno virtual de aprendizaje de la universidad. Con la solución del presente diploma se podrá tener acceso, sin afectar el trabajo diario del entorno virtual, a una cantidad de información relevante para dirigir el proceso docente educativo. La estructura de los datos que se obtuvo en la presente propuesta, permite aplicar algoritmos de minería de datos que conlleven a tomar mejores decisiones y a la larga, estarán incidiendo con unos de los objetivos de la institución: formar un profesional competente en la rama de las ciencias informáticas.

Y para que así conste, se firma la presente a los 10 días del mes de Junio del año 2014

Ing. Eduardo Alfonso Sánchez	Profesor.
Representante de la entidad	Cargo
Firma	Cuño

Glosario de términos

MD: mercado de datos.

ETL: proceso de extracción, transformación y carga.

BI: inteligencia del negocio.

UML: lenguaje visual para especificar, construir y documentar un sistema de software. Sus siglas vienen dadas por su nombre en inglés *Unified Modeling Language*.

XML: estándar de información cuyas siglas vienen dadas por su nombre en inglés *Extensible Markup Language*.

DB2: gestor de bases de datos relacional.

SQL: lenguaje de consulta estructurado o SQL (por sus siglas en inglés *Structured Query Language*). Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

JDBC: es el acrónimo de *Java Database Connectivity*, una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

Base de datos relacional: es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer relaciones entre los datos (que están guardados en tablas), y a través de ellas relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional".

DATEC: Centro de Tecnologías de Gestión de Datos.

No conformidad: defecto, error o sugerencia que se le hace al equipo de desarrollo una vez encontrada alguna dificultad en lo que se está evaluando.